Technical Report

urn:nbn:de:gbv:830–tubdok–10856

Indoor Navigation and Location Based Services Scenario for Airports

Helge Klimek, Björn Greßmann and Volker Turau
Institute of Telematics
Hamburg University of Technology
Hamburg, Germany

{helge.klimek,gressmann,turau}@tu-harburg.de

February 2011

Institute of Telematics
Hamburg University of Technology
TUHH

Table of Contents

Chapter 1

Introduction

This report describes usage scenarios of the architecture for *indoor navigation and location based services* (INLBS) developed in the Airport2030 project. The expandable architecture offers a framework and predefined services with a runtime environment and aims at enabling ubiquitous INLBS in any building with the users' mobile device.

The architecture consists of a *mobile user platform* (MUP) installed on mobile devices and a *stationary facility platform* (SFP) which is running on backend servers. For communication between MUP and SFP a *communication and positioning infrastructure* (CPI) is used. The SFP abstracts from different mobile hardware and communication and ranging technologies for ease of development. A runtime environment contains the service container which invokes life cycle functionality of the different services. Applications which are accessed from the MUP are developed using the services in the service container.

The rest of this document is structured as follows: In the next chapter basic user services will be explained. That chapter is followed by a description of an example setup using two different CPIs. Chapter 4 presents a scenario for using INLBS at an airport terminal.

Chapter 2

User Services

The SFP hosts different kinds of services: runtime services and user services (see Figure 2.1). Runtime services cannot be undeployed from the SFP. Together with other building blocks they encapsulate essential functionality of the SFP. Consequently, they have to be present in all SFPs. User services are services which can be deployed and undeployed at runtime by an administrator. Application developers implement user services which use or extend runtime or other user services in order to implement own applications for a particular SFP. To deploy a user service on another SFP, dependent services have to be present there. A consequence of user services being able to be undeployed is, that runtime services cannot expect them to be present.

Rule 1

Runtime services must not rely on user services.

Services are the building blocks for applications running in the SFP. Composing services in the application development process fosters code reuse and speeds up the development. The following user services are essential for providing INLBS services, but they are not necessary for the SFP to provide its functionality. In certain cases, it might be necessary to undeploy these services or restrict access to them.

Location

The *location* service is used for geocoding. It resolves geographic coordinates to human understandable entities such as room or hallway names. Vice versa, it resolves names to geographic coordinates or polygons. It is important to note, that this service has to be robust

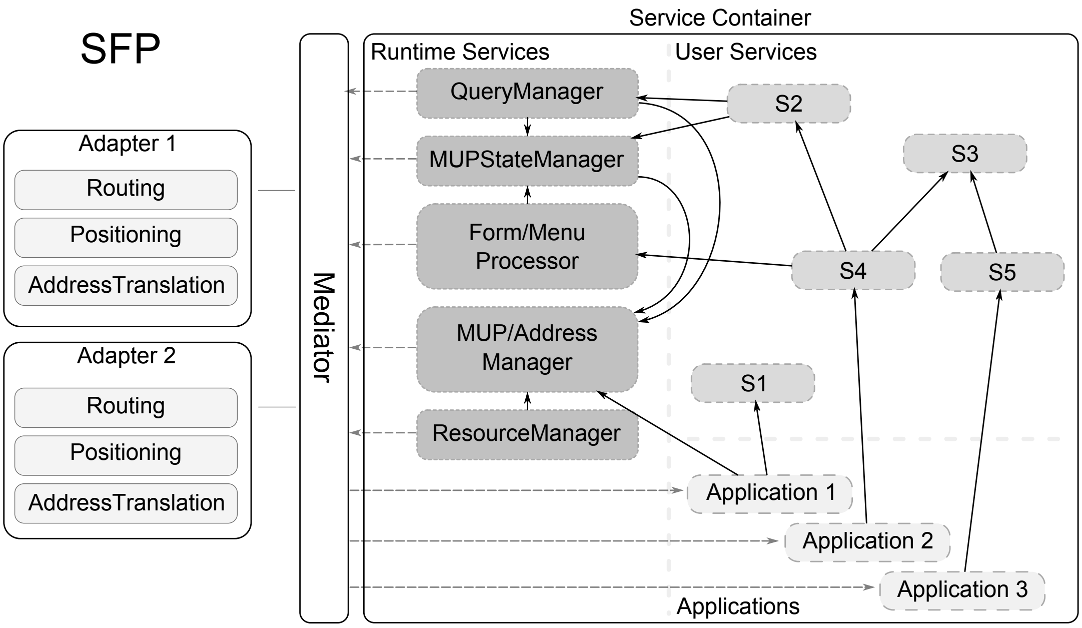

Figure 2.1: Schematic of Stationary Facility Platform.

concerning the naming. User input is bound to natural language, thus different descriptions can denote the same position, e.g. "1st Floor, room 104", "R1.104" and "Konrad Zuse Room".

Notification

The *notification* service allows to register actions with events. Actions will be invoked once the event is triggered. After registering, the service returns a handle which allows to unregister from the notification service, check its state, pause and resume it or request to persist it.

Messaging

The *messaging* service allows to send messages to different entities (not only people). It is possible to send messages to users, geographic regions or all known devices, e.g. emergency broadcast messages.

Navigation

The *navigation* service provides users with a path to a chosen destination. This services requires information about the topological structure of the building to be able to calculate routes. To translate user input into geographical positions, the location service is used. Depending on the capabilities of the device, different representation formats for the guidance process may be selected.

Chapter 3

Communication and Positioning Infrastructure

In Figure 2.1 multiple adapters are shown. Each adapter is connected to a communication and positioning infrastructure (CPI). A SFP has to be connected to at least one CPI through an adapter which hides the technology and hardware dependent properties of the communication and ranging hardware from the SFP. A SFP system can use various CPIs to support different communication and positioning technologies. Each of the different CPIs is connected to the SFP via an adapter. The mediator holds a table which maps communication addresses to adapters in order to route data from the SFP to any MUP over the correct adapter.

In the following an example setup for an airport terminal using two different CPIs is given and the abstraction provided by the adapters is illustrated. An airport terminal could be equipped with a WLAN infrastructure to support smart phones and laptops of passengers and an IEEE 802.15.4 based infrastructure.

The first CPI, the WLAN CPI, is installed using a number of access points deployed in the terminal and using IEEE 802.15.11 b/g/n to communicate with the MUP installed on smart phones, tablets and laptops. The localization of the MUPs is cell based, position estimates are calculated in the adapter based on which access points one able to communicate with the mobile device.

The second CPI, the IEEE 802.15.4 CPI, is also based on a number of base stations installed in the terminal. The MUPs using this CPI are frequent flyer cards equipped with a IEEE 802.15.4 transceiver the airport gave to VIP customers. Localization is based on the IEEE 802.15.4a standard and position estimates are calculated on the device.

Data from any IEEE 802.15.4 or WLAN MUP is sent to a base station or access point and is transmitted to the according adapter. The data is encoded and forwarded to the mediator. To send data from the SFP to a MUP, the mediator looks up the adapter by checking the communication address of the destination MUP and forwards the data to the according adapter. If the data is destined to a IEEE 802.15.4 card, the IEEE 802.15.4 adapter is selected. The adapter translates the communication address into the hardware dependent address of the frequent flyer card and looks up the nearest base station for the MUP based on which base stations were used for recent message exchange and the devices position.

If positions of the MUPs are requested by the SFP, the position is provided by the according adapter abstracting from the hardware details and localization algorithm. While the IEEE 802.15.4 adapter may first send a request to the frequent flyer card before replying the position of the target device, the WLAN adapter can reply a position estimate directly.

Chapter 4

Example: INLBS in an Airport Terminal

This example describes a scenario where INLBS are used in an airport terminal. A user wants to navigate inside an airport terminal to the designated departure gate using a smartphone. The MUP software has been downloaded and installed previously. Upon entering the terminal the MUP detects the SFP by receiving an announcement message. In the following handshake phase, the MUP sends a communication address request with its device ID to the SFP. This request is intercepted at the adapter of the corresponding CPI. The routing mechanism of this CPI and the way the request arrives at the adapter are dependent on the CPI. The adapter buffers the unique device ID and sends a request for a new communication address to the SFP. The mediator dispatches this request to the MUP/address manager which replies with a communication address. At the adapter the new communication address is mapped to the previously buffered device ID. Then a communication address reply is sent to the MUP in order to signal the success of the request. The MUP does not need to know its communication address as the mapping between communication address and hardware is performed at the adapter. Now, the MUP sends its profile.

The MUP starts sending heartbeat messages and the SFP opens a session for the user. In this example, an authorization is not necessary, because the SFPs applications are available to the public. Initially, the menu on the MUP is empty. The MUP sends a request with the root identifier $ID = 0$ to the SFP. In this example, the SFP replies with two menu items: One which will invoke the navigation and one which opens a submenu for flight schedule information (see Figure 4.1).

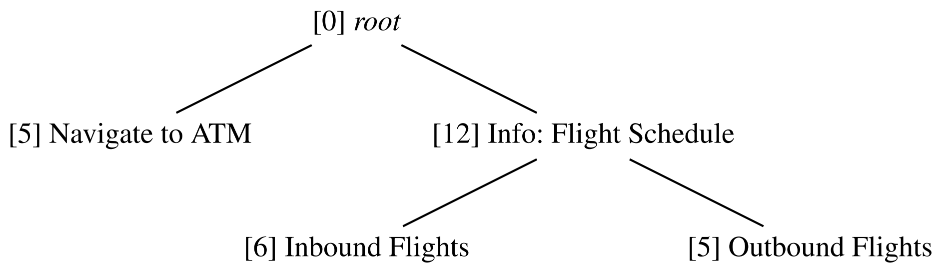

Figure 4.1: Menu tree loaded by the MUP.

The mediator dispatches the initial request to the menu/form processor which holds the information of the menu tree and calculates the reply. The SFP replies with a list of menu items: One with $ID = 5$, which is labeled "Navigate to ATM" and with the description "Select automated teller machine and navigate to it". The other menu item has the $ID = 12$ and is labeled "Info: Flight Schedule" with the description "Get the latest flight schedule". When the MUP receives the reply, the reply is dispatched to the menu engine and the menu items are integrated into the graphical user interface.

In the next step, the user selects the "Info: Flight Schedule" entry. The MUP sends a request with the $ID = 12$ of the menu item to the SFP, and receives a reply with two menu items: The first menu item contains the $ID = 6$ and label "Inbound Flights" and description "List of Inbound Flights". The second menu item has the $ID = 7$ and is labeled "Outbound Flights", its description is "List of Outbound Flights".

The user selects the "Outbound Flights" item and the MUP sends a request with $ID = 7$ to the SFP. The menu/form processor determines that the child element of the selected item is a form with a list containing the outbound flight numbers and their departure time. The SFP sends the form which already contains default values and its description id to the MUP. The interpreter receives the form and checks the form cache to determine whether the form description has been loaded before (see Figure 4.2). As this is not the case, the MUP requests the form description from the SFP, and caches it once it is received.

The user goes back in the menu and selects the navigation item. The MUP sends a request with $ID = 5$ to the SFP and the menu/form processor in the SFP determines that the child element of the selected menu item is a form. This form is sent to the MUP. After receiving the form description and caching it, the form is rendered by the MUP on the mobile device. The user fills in the form, by selecting an ATM from a list of destinations which have been sent to the MUP and clicks on the submit button. The form data is sent back to the SFP and the mediator dispatches the form data to the Form/menu processor and finally to the navigation application.

The application queries the mobile device for sending its position data periodically using the query mechanism. Queries to the MUP which do not originate from the owner have to be confirmed for privacy reasons. In this example, a confirmation is not necessary since the user started the navigation. In a possible different case, where position data is consumed by other users, the user would have to give his consent to the application to process the position data. Since the MUP has not already allowed other applications to use its position data and is not sending it periodically, it starts doing so now.

When the SFP receives the position data, it dispatches the data packets to the application which invokes the navigation user service. The service calculates route updates and navigation instructions and checks whether the user has reached its destination. These navigation updates are pushed to the MUP via the asynchronous resource push mechanism. The specific resources that are pushed to a MUP depend on its profile. Depending on the graphical capabilities of the device, navigation instructions can be sent in various ways, e.g. textual messages, direction arrows or paths on maps.

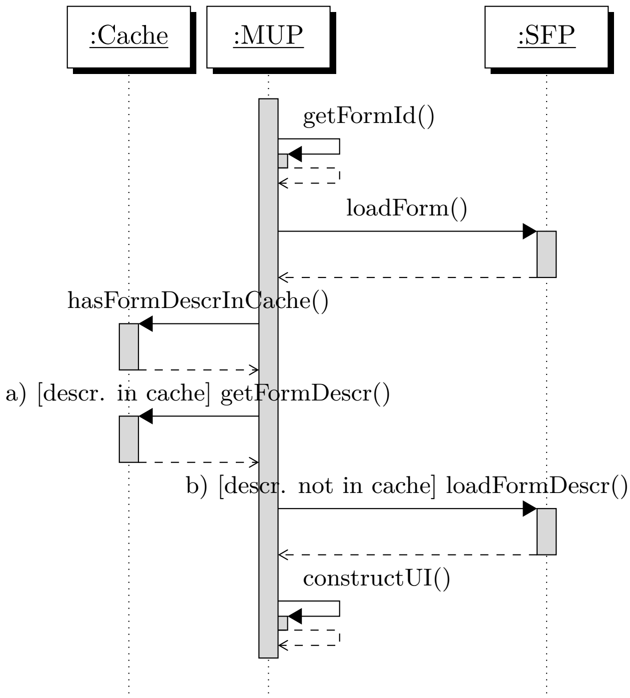

Figure 4.2: Accessing forms from SFP or cache.

New navigation steps are pushed to the MUP until the user has reached the destination. If no other authorized applications are enlisted in the SFP, the MUP will stop sending position updates.

In order to prepare meals for the flight, airline staff invokes a user service on the SFP to request the information whether a passenger is a vegetarian. This information is stored as a property on the passengers MUP. The SFP user service thus invokes the query manager runtime service. A query for this property is sent from the query manager to the MUP. At the MUP the query is dispatched to the query management which reads the query and looks up the property in the property store. The property is sent back to the SFP, dispatched to the query manager and the invocation of the query manager runtime service is terminated with returning the property to the requesting service. If the request for the property originates from another user, the MUP user is asked to confirm the request.