Deploying Contextual Computing in a Campus Setting

Fabio Aversente, David Klein, Schekeb Sultani, Dmitri Vronski, Jörg Schäfer
Department of Computer Science
Frankfurt University of Applied Sciences
Frankfurt am Main, Germany
Email: {faversente, d.klein, schekeb.sultani, dmitri.vronski, jschaefer}@fb2.fra-uas.de

***Abstract*—Location Based Services (LBS) are regarded as a major constituent of contextual computing. However, deploying LBS for indoor localization remains still a largely unsolved problem, in particular if practical considerations like, e.g., effortless calibration are taken into account. In our work we analyze challenges in a university environment characterized by hundreds of access points deployed and by heterogeneous mobile handsets of unknown technical specifications and quality. We developed an open architecture to deploy LBS on a campus and integrate them with other services and useful applications to support campus life.**

***Index Terms*—Location Based Services, Indoor Localization, Contextual Computing, Architecture, Mobile**

I. Introduction

A. Contextual Computing

The vision of contextual computing has been around more than two decades and with the ubiquitous availability of connected smartphones and similar devices it is slowly becoming reality. According to the vision of Marc Weiser "The best computer is a quiet, invisible servant" [1], the contextual information should be obtained automatically with no or minimal user interaction. The goal is to use different contexts (e.g., location, time, ID) to improve forecasts and context-sensitive services. One particularly important aspect of contextual computing is the awareness of a user's location. In this paper we describe practical aspects of deploying an infrastructure for contextual location awareness in a university's campus setting for indoor environments.

B. Indoor Localization Overview

For an overview on indoor localization based on WiFi, we refer to [2]. In general, techniques either make use of triangulation based on a theoretical propagation model or on scene-analysis, i.e., fingerprinting. Although propagation models look attractive from first principles, the lack of a good theoretical foundation for indoor radio propagation yields less than optimal results in the field. Henceforth, most applications use fingerprinting, where the signal is matched (testing phase) to previously recorded signals (training phase) with some probabilistic model. One of the earliest such systems is the Horus system, see [3], for a general overview of probabilistic techniques see, e.g., [4]. With these techniques one can achieve localization accuracy of 2-3 meters and 95% room classification (depending on the room topology) in areas of high coverage of WiFi signals, i.e., with many accessible access points for fingerprinting, see [2] and [5]. The disadvantage of these methods is the relatively high effort for the training phase which requires a careful measurement that needs to be recalibrated whenever a change of the topology happens. Furthermore, in practical applications the testing data is usually generated by a different handset than the ones used for training which further complicates the analysis and reduces the accuracy. Our approach tries to minimize these efforts by choosing robust classification techniques [5] and generating training data (semi-) automatically with the help of crowdsourcing, see Section II-C1.

II. Offered Services

A. MoCa and CoCo Project

In 2012 at Frankfurt University of Applied Sciences the project MoCa (Mobile Campus Applications) was launched. Its mission is to provide personalized, context-sensitive services to students (and university staff) based on individual and role-based requirements. To provide context information for many services such as lecture and seminar support, an indoor localization service was developed as a core component of the MoCa infrastructure. Among the services voting applications (see, e.g., Section IV-A) or social network services such as "find-my-buddy" applications (see, e.g., Section IV-B) are supported. One of the key requirements of the project is (close-to) zero maintenance efforts for the operations of this service due to resource constraints. Henceforth, any time-consuming efforts such as, e.g., expensive calibration must be avoided and we rather try to make use of crowdsourcing techniques to facilitate the data generation (training) process. For a comprehensive overview of the MoCa vision we refer to [6]. In 2015 the Project Contextual Computing (CoCo) was launched to investigate how contextual information could be used to develop intelligent personalized applications. The main focus was initially on indoor localization techniques. It is currently investigated how to improve the accuracy and/or performance of the existing localization approach developed by the MoCa Project. To achieve the goals of contextual computing, we need to aggregate data from several sources. As the project aims to enhance the campus life at Frankfurt University of Applied Sciences, the data can be broadly categorized as data related to the students' studies, location based data and general campus related data. For this purpose, machine learning methods such as particle filters [7] are applied and data from different sensors are fused.

B. Student Lifecycle Management

As the most relevant information for students using the services relate to their studies, namely their courses, grades, and general information about the university, the project integrates with the Digital Campus (DC) application of the university. The DC project is based on the Student Lifecycle Management (SLCM) system from SAP and provides a portal for students to keep track of their studies. It also allows the retrieval of information required for the organization of the program, e.g., the timetables and room occupations. For further background information we refer to [8].

C. Location Based Services (LBS)

To avoid unnecessary distractions we aim to filter the presented data to only currently relevant services. Besides the user's identity, one of the main indicators for relevance is the user's current location. To provide personalized services to students, distinct services could be available in specific rooms at predefined times only. Thus the applications need to be able to depend on the user's location as they aim to provide a good user experience in the spirit of Marc Weiser [1]. The main source of location information is currently based on WiFi fingerprinting. For a detailed discussion of the used approach we refer to [5].

1) Calibration: The procedure to collect the signal strength values for the initial calibration (described in [5]) was very thorough, involving for example geodetic surveys, and thus very time consuming. While this level of accuracy was necessary for the development and validation of the used techniques, performing similar measurements across the whole campus is not feasible.

An additional issue is that the set of access points is in constant flux as access points are frequently replaced, moved, and new ones installed without or with little notice. Recalibrating the system therefore has to be reasonably simple and autonomous, e.g., via constant recalibration based on some form of crowdsourced data. To this end, the SmartClick application (described in Section IV-A) enables the collection of crowdsourced data in a natural and non-intrusive manner.

2) Extensibility: Our research group aims to refine the localization accuracy and therefore actively participates and supports research to improve the classification results. A recently concluded project attempted to use sensors built into modern smartphones to devise additional approaches to determine the location in a building. In [9] magnetometers were used to measure magnetic field vectors. These provide patterns which allow to calculate the position of the device inside the building. The first results look promising and are currently investigated as part of the project CoCo.

D. External Services

In addition to the services mentioned above we want to provide additional information relevant to the campus life. Many of these are not under control of the universities IT department, contain semi- or even unstructured data, and henceforth different techniques to integrate them into the ecosystem (e.g., web scraping) have to be used. A popular example is the menu of the refectory on the campus, which naturally is of interest for students, see Section IV-C.

III. Architecture and Infrastructure

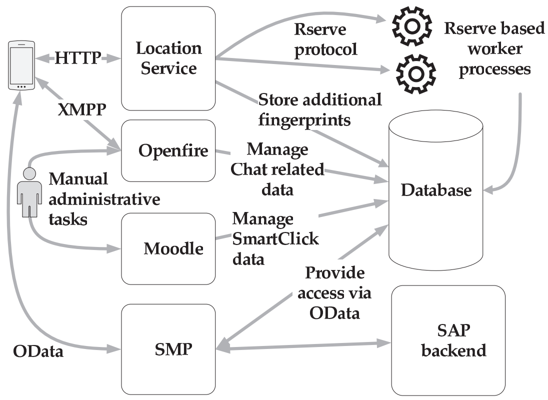


Fig. 1. Moca Architecture – Interaction between components

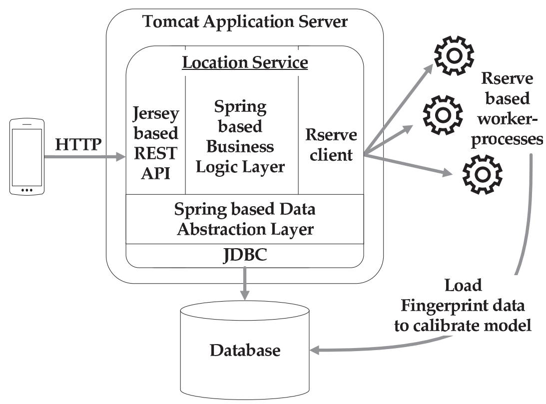


Fig. 2. Location Service – Design Overview

A core ingredient of the MoCa infrastructure is the SAP Mobile Platform (SAP Mobile Platform SDK V. 3.10, SAP Mobile Platform Server V. 3.0.10.0 and Apache Cordova V. 5.1.1), representing the foundation for our mobile applications. In addition we also maintain several specialized services, which provide specific features for (planned) applications.

For all services other than the SAP Mobile Platform we follow an open source and open protocol strategy. For example, the Location Service which performs the fingerprint classification, is build with Java and R, see below. Another example is the chat functionality of the *SmartBuddy* application

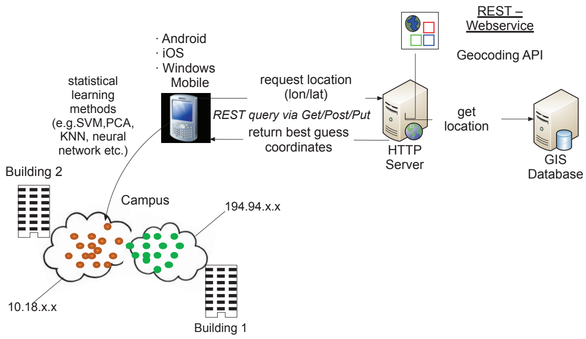


Fig. 3. The localization process

see Section IV. The chat is based on the *Extensible Messaging and Presence Protocol* (XMPP) provided by OpenFire.

Figures 1 and 2 provide an overview on MoCa's overall architecture and design. The details of the data-flow requesting the current position are depicted in Figure 3.

A. SAP Mobile Platform

To make the DC data available to students on their mobile devices, SAP offers the SAP Mobile Platform (SMP) providing the following services:

1) Model driven data access to SAP backend of DC
2) Model driven data access to other backends
3) Model driven data transformation
4) Device registration
5) User authentication and authorization
6) App provisioning

On the server side, SMP offers an unified interface for the SAP based backend infrastructure. We make use of a web service access to the room management service provided by the DC which provides data such as a lecture to room (and lecturer) mapping for each semester. This association is then used by other services such as, e.g., the *SmartClick* Application, see Section IV-A below. Clients can leverage *Open Data Protocol* (OData) based RESTful web services to retrieve and publish information. As OData is an open standard [10] and based on regular HTTP, it is a good fit to our use case.

To develop mobile applications interacting with SMP, we make use of their support for Apache Cordova based Hybrid Web Container applications. These provide a runtime which manages application updates, provisioning, user management, etc.

On top of the SMP runtime mobile applications are developed using web technologies, i.e., JavaScript, and HTML5. This allows us to minimize native developments and therefore helps to develop mostly platform independent applications, available on devices running iOS, Android, Windows Phone and BlackBerry. This reduces the development – and more importantly – the operations costs. Henceforth, the SMP architecture [11] and our development approach [12] allow us to support all major mobile platforms in an efficient manner.

However, for certain services access to device specific interfaces is required. For example, we need to access the WiFi radio to scan for available access points. In order to support this requirement, we developed a native library and exposed it to the Web Container. Thus, native, device- or OS-specific development is minimized and confined.

In addition to the mobile platforms, it is also easily possible to test and deploy the SMP based application as a website. When running in the browser it is of course impossible to make use of native libraries and some of the more advanced SMP runtime features. Nonetheless, the major functionality can be seamlessly tested in this manner.

Overall, due to using the SAP SMP framework, we are able to develop applications targeting all major mobile operating systems while keeping platform specific code to an absolute minimum. Platform specific testing is obviously still required and inherent restrictions make complete feature parity impossible. It nevertheless allows us to keep a unified release schedule, while offering applications with mostly identical feature sets and user interfaces.

B. Location Service

As it is not feasible to perform the classification locally on the user's phone, we chose a centralized approach. This minimizes the work done on the client device and thus optimizes its battery usage. It also avoids the need to distribute and update the data sets required to perform the calculations.

We therefore decided to make the Location Service available through a stateless REST service. To decouple it from the largely unrelated SAP infrastructure, we decided to provide a simple REST web service instead of making it available via OData and SMP.

This has several advantages for us. It allows students to access the service for arbitrary projects, without having to get familiar with OData. The service can also be made available to a wider audience via ad-hoc services, e.g., to use it during external events, like a conference.

The model used to classify the fingerprints and the surrounding functionality is implemented in R. This choice is based on the advantages of an interpreted language suitable for agile methodologies in a research context and the immense popularity of the R language and its ecosystem. With "more than 2 million users" [13] and several thousand open source packages maintained by the community [14], R offers support for most machine learning and data analysis tasks.

In addition, R offers straightforward interoperability with other languages, mainly via bindings for native, performance optimized libraries implementing the underlying functionality. This offsets most performance losses that incur from using R instead of a compiled language like, e.g., C++ or Fortran.

Communication between the REST web service and the R application is realized with the help of Rserve. It acts as a server, making the R process available to clients, allowing them to remotely execute R commands and retrieve the results. Rserve was originally introduced in [15] and is widely used, e.g., in products like SAP HANA [16].

This approach allows us to have a simple Java based REST web service acting as the front-end. It maintains a list of established connections to Rserve processes and forwards requests to said worker processes which perform the actual classification.

As Rserve communicates over TCP/IP, worker nodes can be distributed across several servers. This allows us to distribute the computationally expensive operations across several machines while keeping the overall architecture simple. Using the stateless REST web service as a front-end thus allows us to adjust the number of worker nodes according to user demand without service interruptions for the clients. This also allows us to seamlessly change the underlying calculations, e.g., to refine the classification algorithm, without changes to client applications.

To determine their current location, clients send WiFi fingerprints to our service. Said fingerprints contain information about all access points in range, namely their *basic service set identification* (BSSID), *service set identification* (SSID) as well as the measured *signal strength* (RSS) values. In addition to performing the classification, the service stores the fingerprint data and calculated location. No personal information is recorded whatsoever, i.e., the usage of the service is anonymous.

As long as the installed access points only change gradually, e.g., broken ones are replaced instead of abrupt, larger changes (say, all access points are upgraded to newer hardware), we should be able to continuously recalibrate the system based on the growing data set. This allows us to automatically recognize and incorporate changes in the WiFi infrastructure and adjust the model accordingly. This also avoids the manual recalibration of the system. In the past 24 months the university has replaced roughly 30% of its access points in the computer science building. Although the system was not fully deployed in its current stage at the time, the system could be operated as if it was already automated and we could verify that it would cope with the changes without loosing accuracy. As described in [5], our systems using Support Vector Machines (SVM) for classification yields an average classification error of better than 95% and a mean error of less than $2.5 - 3.0$ meters which is compatible with best results published in the literature, see [5] for details.

To further improve localization accuracy, state dependent methods can help, i.e., methods that do not just analyze the current signals but rather the history of sensor and location data. Sequential Monte Carlo methods such as particle filtering enable to solve the otherwise intractable bayesian filter recursions. An example of a simulation is shown in Figure 4. It depicts a random walk of a potential user of our system, where the dots on the connected line depict the walk and the free (blue) dots represent the particles. The particles approximating the actual user position are clustered close to the right position, i.e., close to ground truth. The errors are comparable to the results obtained via SVM described in the previous section, however, particle filters can fuse other sensor data in a more meaningful manner. Research is currently in progress to further improve the localization accuracy. One approach is to use the log-normal shadowing model which represents the signal strength decrease dependent on the distance.

Another approach is to assess the viability of using the strength of the magnetic field vector of the earth's magnetic field as well as data from other (inertial) sensors (accelerometer, gyroscope) and combine this information with WiFi signals.

Recently [17] it was shown that combining (inertial) sensors with WiFi data using sensor fusion with Kalman filters can further improve WiFi accuracy by as much as 0.5 meters. Thus, for the future we plan to combine this approach with our existing WiFi based location techniques using SVM and particle filters.

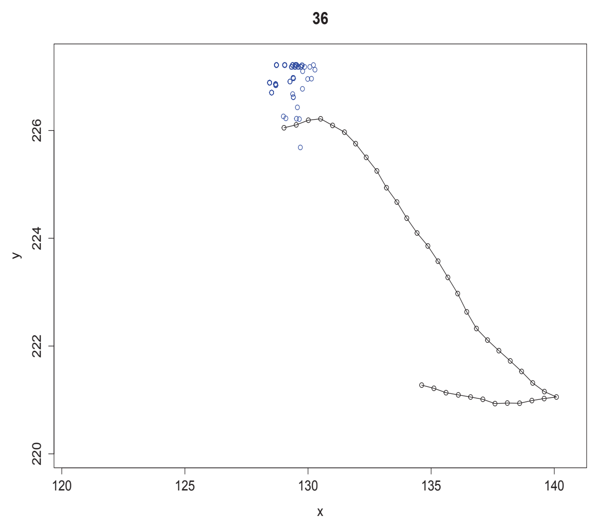


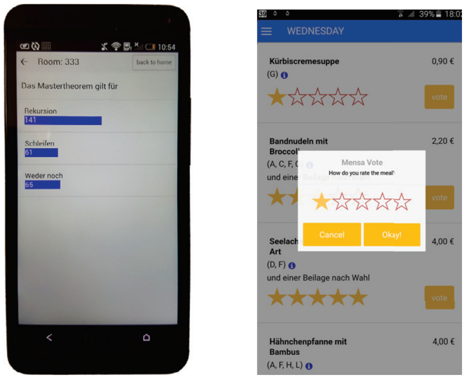
Fig. 4. Particle Filter

IV. Applications

A. SmartClick

SmartClick is an audience response system (ARS) developed for classroom usage. It helps lecturers to keep track of how well students understand the study material at hand by taking short surveys during the lectures.

By incorporating contextual information, we aim to minimize the overhead caused by using non specialized hardware, namely the students' smartphones, as responders. The application uses the current date, time, and the users location to automatically determine which lecture the user currently attends. This allows us to only display questions relevant to the student. The automatic preselection minimizes the user interaction required to participate in the quiz and consequentially its duration. Results are available both in the mobile application (Fig. 5a) and on the courses website. The mobile application also allows to correct wrong localization results by manually selecting a room. The corrected data point can then be sent back to



(a) SmartClick (b) SmartMensa

Fig. 5. SmartClick and SmartMensa App

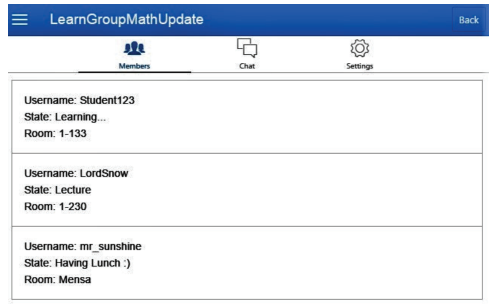


Fig. 6. SmartBuddy App – The current locations of your friends

Fig. 7. SmartBuddy App – Chat functionality

the server, allowing us to crowdsource the collection of new measurements.

SmartClick also provides a web interface. Lecturers use the website to create and maintain questions while students can access results of already concluded polls, allowing them to revisit past topics. The web interface is implemented as a plugin for the Moodle [18] learning platform. Moodle is a widely used online learning platform, providing access to course related information and materials over the Internet. Integrating our application into the e-learning platform ensures that the course specific information (e.g., time, location, start date, and end date) used by our system is available and up to date, avoiding any manual maintenance. This also applies to user account management. All students as well as staff members are registered on the Moodle platform and use it to sign up for lectures they attend. As a plugin, our application automatically uses the course and user information already maintained in Moodle as well as its security infrastructure. This architecture additionally ensures that potential users are already familiar with the application and its user interface.

Integrating parts of our infrastructure into existing and widely used systems, here the online learning platform, helps with minimizing the effort required to introduce as well as to maintain an audience response system in classroom settings.

B. SmartBuddy

SmartBuddy (Figures 6 and 7) is a “find-my-buddy” application locating “buddies” automatically on the campus. Aimed at students, it helps to organize study groups and keeping in touch with other students without the necessity to exchange personal information (e.g., mobile phone number, email address). Students need to create an account to use the application. Once logged in they can create and join groups. To offer more privacy, groups can be password protected. Existing members then have to explicitly invite interested users by sharing the secret password allowing to join the group. Each group has an associated chat room. As the applications intends to facilitate studying in groups, it also allows users to share their current location. This can be used by study groups, helping with the coordination of (spontaneous) encounters on the campus.

C. SmartMensa

SmartMensa is an application for students and staff who intend to eat at the university refectory. It displays the current menu, together with recommendations. The ratings are crowdsourced, collected from other users who use the application (Fig. 5b) to anonymously rate the food.

Whilst mainly intended for fun, it serves as an opportunity to experiment with and test the infrastructure. It is useful for all people frequenting the campus. It therefore also helps to bring more users into our application ecosystem and to generate additional feedback.

V. Operations

The Location Service went live in early 2015. During the operations period, several changes to the access point infrastructure took place. Figure 8 shows the results of cross validating the data of six selected handsets partitioned into 15 randomly chosen subsets for training and validation versus training data of 60 combinations of only two handsets and tested against a third one. The statistical aggregates of the error measures as described in [5] are depicted in Table I. As one concludes from the tabular data, using only two handsets yields a much worse median classification error as compared to training data with 6 handsets. Furthermore, the variance as depicted in Figure 8 is *much* higher and renders some tests practically useless (median error bigger than 50%). This demonstrates one of the practical difficulties in training the system. In our case the major contribution to the data being worse than the one collected in [5] stems from two particular devices that show very different RSS characteristics compared to the others. We found empirically that training with at least half a dozen devices from different OEMs is advisable to avoid strong dependency on single device types. Recently we have started to use crowdsourcing to collect enough data from a large set of mobiles, thus helping to mitigate device dependency on a continuous basis.

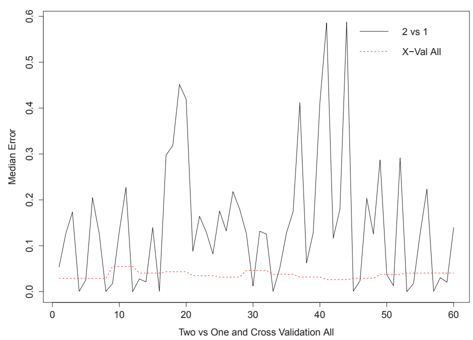


Fig. 8. Two vs. One and Cross Validation

TABLE I
PERFORMANCE – TWO VS. ONE AND CROSS VALIDATION ALL

	median(e)	max(e)	median(m)	min(p)	min(r)	min(f)
2 vs 1	0.145	0.177	NA	0.694	0.446	0.626
Cross-Val	0.038	0.052	NA	0.857	0.744	0.835

As we expand the coverage of the rooms, we noticed another phenomenon, namely the incorrect classification of rooms between different *floors*, see, e.g., the confusion matrix, Table II mixing rooms 131 (first floor) and 401 (fourth floor). This could be the result of the open architecture of the computer science building (e.g., atrium, and galleries). These type of errors are easily taken care of by state-based machine learning techniques such as particle filters investigated in the CoCo project.

TABLE II
CONFUSION MATRIX

	129	130	131	234	235	332	333	401
129	33	0	0	0	0	0	0	0
130	1	38	1	0	0	0	0	0
131	0	0	40	0	0	0	0	0
234	0	0	0	31	0	0	0	0
235	0	0	0	0	25	0	0	0
332	0	0	0	0	0	45	0	0
333	1	0	0	0	0	4	44	0
401	0	0	7	0	0	0	0	75

VI. CONCLUSION

We have demonstrated the feasibility of SMP for agile development and integrating contextual services via REST. We have also shown the practical feasibility of a location based service to support campus services with room accuracy.

For future research we intend to further improve accuracy by better algorithms and sensor fusion. Furthermore, in addition we will integrate sensor data from other sources to improve contextual computing and will make use of reality-mining techniques to improve services.

REFERENCES

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, September 1991.

[2] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Trans. Sys. Man Cyber Part C*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007. [Online]. Available: http://dx.doi.org/10.1109/TSMCC.2007.905750

[3] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '05. New York, NY, USA: ACM, 2005, pp. 205–218. [Online]. Available: http://doi.acm.org/10.1145/1067170.1067193

[4] P. Kontkanen, P. Myllymaki, T. Roos, H. Tirri, K. Valtonen, and H. Wettig, "Topics in probabilistic location estimation in wireless networks," in *Proc. 15th IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications*. Barcelona, Spain: IEEE Press, 2004. [Online]. Available: http://cosco.hiit.fi/Articles/pimrc2004.pdf

[5] J. Schäfer, "Practical Concerns of Implementing Machine Learning Algorithms for W-LAN Location Fingerprinting," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2014 6th International Congress on*, Oct 2014, pp. 310–317.

[6] B. Jaser, "Design and set-up of an architecture for the development of a framework for location-based, mobile campus applications," Master Thesis, FH-Frankfurt, Jan. 2014.

[7] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[8] C. T. Ulrich Schrader, *Best Practice to IT : erste Erfahrungen aus dem Projekt Digitaler Campus*. DHI Deutsches Hochschul-Institut, 2012, vol. Schriftenreihe Hochschulen im Fokus, no. 5.

[9] A. Steinbrecher, "Indoor localization using magnetic field vectors," Master's thesis, Frankfurt University of Applied Sciences, June 2015.

[10] OData Technical Committee. OASIS open data protocol (OData) TC. Accessed May 17, 2016. [Online]. Available: https://www.oasis-open.org/committees/odata/

[11] *SAP Mobile Platform Version 2.3 – Architecture*, 2013, accessed May 17, 2016. [Online]. Available: http://scn.sap.com/docs/DOC-42440

[12] *Sybase Unwired Platform 2.1.x Development Paradigm White Paper*, 2012, accessed May 17, 2016. [Online]. Available: http://scn.sap.com/docs/DOC-30196

[13] Revolution Analytics, "The R community," accessed May 17, 2016. [Online]. Available: http://www.revolutionanalytics.com/r-community

[14] C. R. Maintainers, "The comprehensive R archive network," accessed May 17, 2016. [Online]. Available: https://cran.r-project.org/

[15] S. Urbanek, "Rserve – a fast way to provide R functionality to applications," in *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, 2003.

[16] *SAP HANA R Integration Guide*, 1st ed., May 2015, accessed May 17, 2016. [Online]. Available: http://scn.sap.com/docs/DOC-60314

[17] C. Bianucci, "Enhancement of wifi positioning with a low-cost inertial measurement unit," Master Thesis, Frankfurt University of Applied Sciences, Jan. 2016.

[18] Moodle Docs 3.0. Accessed May 17, 2016. [Online]. Available: https://docs.moodle.org/30/en/Main_page