Integrated Scenario for Machine-Aided Inventory Using Ambient Sensors

Timo Schairer, Christian Weiss, Philipp Vorst, Jürgen Sommer, Christian Hoene, Wolfgang Rosenstiel, Wolfgang Straßer, Andreas Zell, Georg Carle. Computer Science Department, University of Tübingen, Tübingen, Germany
Patrick Schneider, Anette Weisbecker. IAT, University of Stuttgart, Stuttgart, Germany

Abstract

We present a novel complete system for machine-aided inventory. Our system covers automatic product identification using RFID, localization based on ambient sensors, the enrichment of raw RFID data with product information from ERP (Enterprise Resource Planning) backend systems and real-time augmented reality (AR) visualization. We have integrated all of these components into a real-world demonstrator resembling a supermarket and successfully presented the system in the scope of an industry workshop.

1 Introduction

In order to simplify processes in logistics, warehousing, and surveillance, our interdisciplinary joint project AmbiSense [1] has combined solutions for efficient acquisition and mapping of environments. These environments are equipped with diverse ambient technology such as WLAN, Bluetooth, and RFID. The research is based on techniques stemming from the fields of robotics, embedded systems, AR and ERP.

This paper is organized as follows: Firstly, we define our application scenario. Then, Section 2 gives an overview of the complete system architecture, followed by an in-depth description of the different components of our system. We present our results in Section 3 and finally draw a conclusion in Section 4.

1.1 Application Scenario

One key component of our project is the continuous integration of all developed algorithms and techniques into a real-world demonstrator to illustrate their practicability and usefulness. We have chosen warehousing and retail as our current application scenario: Robot-assisted inventory is applied in a supermarket as we expect goods to be labeled individually with RFID tags in the near future. This enables products to be tracked from production to sale consistently and to be localized permanently. In order to provide a working demonstrator, we set up an application scenario resembling a supermarket at the AmbiSense lab at the University of Tübingen. It consists of individually tagged products placed in typical shop shelves.

Our robot, equipped with an RFID reader, traverses the supermarket environment while constantly detecting products within its range. The data are transmitted using WLAN to a central computer which holds a model of the current state of the system. We augment these data by additional product-specific information provided by the ERP system. The detected objects as well as additional product data are visualized using AR techniques. The robot localizes itself using the existing infrastructure of different ambient sensors (RFID, Bluetooth, WLAN).

This scenario aims at synchronizing the product stock of supermarkets or stores automatically. Other sample tasks could be the identification of products that are past their sell-by dates or located in the wrong places.

1.2 Contributions

We present a novel prototype system for robot-assisted inventory targeted at supermarkets. Our system consists of various distributed components for robot control, localization, sensor and data fusion, ERP backend connection and visualization. To provide maximum extensibility, flexibility and reusability of these components, we apply principles of service oriented architectures (SOA) for communication of the distributed system parts.

1.3 Related Work

Related work with respect to our application scenario is most profoundly influenced by the Metro Group, which currently utilizes RFID tags for palette identification [5]. Wal Mart is another company which increasingly focuses on RFID for palette identification. With our work we intend to extend this identification to the product level at the point of sale. Additional features of our approach include autonomous robots and a real-time visualization as well as self-localization of mobile agents (i.e. robots, service staff using PDAs or laptops) using ambient sensors (RFID, Bluetooth, WLAN).

2 Overall System

The AmbiSense system consists of four distinct parts that are described in more detail in the following subsections and depicted in Figure 1.

The first part is composed of mobile platforms, typically robots or handheld devices. These mobile agents constantly locate themselves and gather data from their surrounding, such as images and RFID tag data (Sections 2.1 and 2.1).

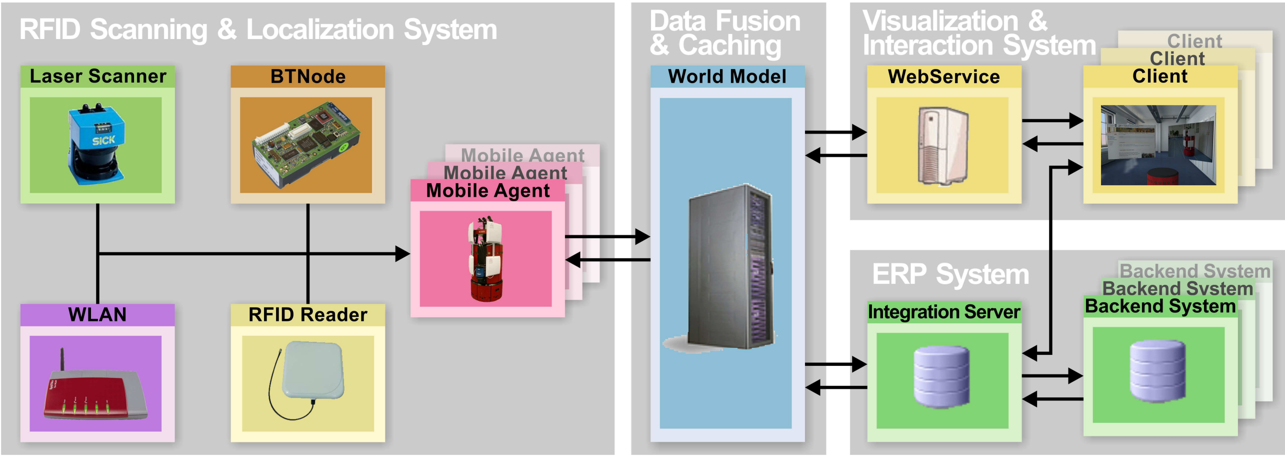

Figure 1 Schematic overview of the overall system.

The data acquired by these mobile agents are typically raw data, for instance the numbers stored on RFID tags.
The mobile agents transfer the gathered data to the second part of the system, the so-called world model (Section 2.3). The world model acts as a central point of integration for all connected components.
The third part of the AmbiSense system is the ERP integration server (Section 2.4), which is responsible for the enrichment of the gathered raw data with more detailed information from the connected backend systems.
In order to enable humans to understand and manipulate the abstract data, the fourth part provides an intuitive visualization (Section 2.5) by means of augmented reality.

2.1 Robots and Localization

As mobile platform, we use an RWI B21 service robot (Figure 2). It possesses a so-called synchro drive, which allows the robot to turn on the spot. Mounted on the front of the robot, there is a SICK LD-OEM1000 laser scanner (240° field of view in our setting). We use the laser scanner for obstacle avoidance and to get ground truth position information. On its top, the robot carries color cameras for object recognition. These cameras are placed on a pan-tilt unit so that the orientation of the cameras can be altered automatically. The main sensor for our scenario is an Alien Technology ALR-8780 UHF RFID reader, which supports the UHF standard EPC Class 1 Generation 2. We fixed the two antenna pairs connected to the reader to the front left and right side of the robot so that the antenna pairs span an angle of about 90°. This setup allows the robot to scan a relatively large area.
Inside its body, the robot carries two standard PCs. Among other tasks, they control the robot's engines, process the sensor data, and compute the current position of the robot. For communication with external servers, for example the world model, the robot is equipped with a WLAN card and two Bluetooth dongles. On its rear side, our robot is additionally endowed with a touch screen monitor. This monitor displays information about recognized products and about the current state of the robot. Furthermore, humans can interact with the robot using this display. This way, we could use the robot as a shopping guide in the future.
A key prerequisite for robot navigation is that the robot knows its current position. In indoor environments, self-localization of mobile robots is commonly based on laser data. However, laser scanners are relatively expensive. In order to save these costs, our goal is to use existing infrastructure for self-localization (e.g. already installed WLAN nodes). Therefore, we developed three alternative localization methods based on RFID, WLAN and Bluetooth. In this paper, we only give brief descriptions of the three methods. A detailed description can be found in [23].
In order to localize the robot using RFID, we place a number of stationary RFID tags in the environment (for example at shop shelves or at walls). We assume that we know the position of each tag. If the robot detects one of these tags, it must be somewhere near the tag. Consequently, the approximate position of the robot is known. However, the exact distance and orientation of the robot with respect to the tag is still unclear. Fortunately, the robot usually detects more than one stationary RFID tag at once. By incorporating the information from multiple detections, a more accurate position estimate is possible. Additionally, the integration of different measurements over time greatly contributes to increasing the localization accuracy.
Currently, we employ two different localization approaches: a method that requires an explicit sensor model of the RFID antennas, and a method based on fingerprinting. Both methods use a so-called particle filter [19], which estimates the robot's position based on the sensor measurements in a probabilistic fashion.
The sensor model-based RFID localization method was originally proposed by Hähnel et al. [13]. This approach uses an explicit sensor model of the RFID antennas to model the probability of detecting an RFID tag at a relative position with respect to the antenna. The generation of the sensor model is very time-consuming, because one must manually

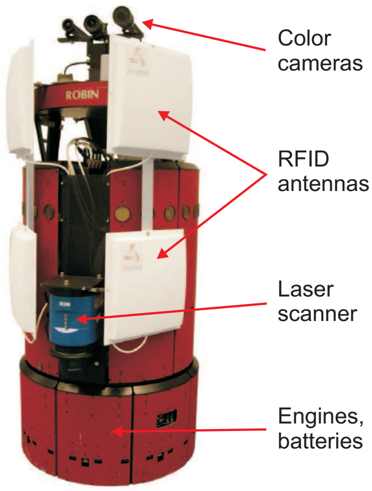

Figure 2 Our service robot and its main sensors.

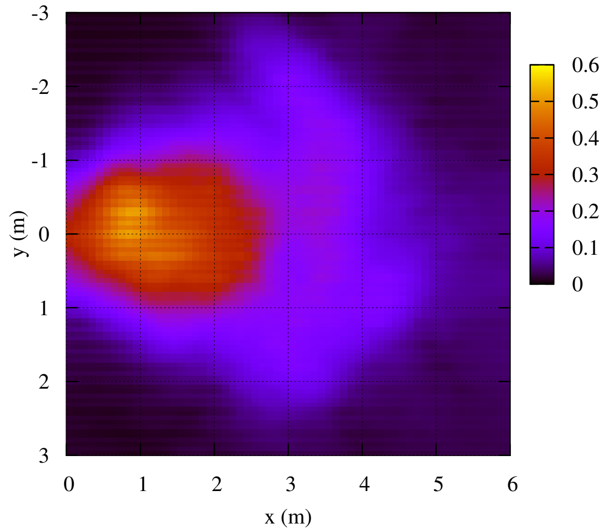

Figure 3 Automatically generated sensor model for our RFID antennas. The antenna is located at the origin $(0, 0)$. The sensor model represents the detection probabilities of RFID tags at different positions relative to the robot.

determine the detection rates of RFID tags for a large number of tag positions relative to the antennas. Therefore, we extended the approach by a technique that automatically learns the sensor model while the robot is exploring the environment [24]. Figure 3 shows such a manually generated sensor model for an RFID antenna. The automatic generation of the sensor model is at least five times faster than the manual generation.

For the fingerprinting-based localization method, the so-called *RFID snapshots* were developed within the scope of the AmbiSense project [17]. This method does not require an explicit sensor model, but learns the distribution of the stationary tags in a training phase. Basically, an RFID snapshot represents a robot pose (position and orientation of the robot) by a vector. This vector stores for each tag its detection probability, given that the robot is located at the specific pose. The map of the environment consists of the set of snapshots recorded in the training phase, together with the poses of the snapshots. Later, the robot localizes itself by comparing the vector of currently detected tags to the stored RFID snapshots.

Our Bluetooth-based self-localization approach is based on *received signal strength indication* (RSSI) to Bluetooth landmarks in the environment [18]. Thereby, signal strength is mapped to distances. A novel feature of our approach is the automatic parametrization of the sensor models. To compute the position of the robot from multiple RSSI values, we examined a trilateration by minimizing the squared errors (MSSE). As an alternative, we use a particle filter. Because many mobile devices like smart phones or PDAs possess a Bluetooth interface, the Bluetooth localization is not restricted to an application on a robot but can be used on a wide range of different devices.

Finally, we also use existing WLAN infrastructure to localize the robot with respect to fixed WLAN nodes. This method is based on round trip times, i.e. the time of flight of signals between WLAN devices [12]. We developed a new algorithm called *Four-way Time-of-Arrival*, which is faster, more accurate and easier to implement than previous approaches. Additionally, the algorithm allows for cooperative localization. Furthermore, we plan to release the open-source software *Goodtry* [14] for WLAN trilateration.

2.2 Vision-based Object Recognition

When the robot navigates through our supermarket scenario, it continuously scans for tagged products with its RFID receiver. As an alternative way to recognize products, we use visual object recognition. Our object detection uses local image features to search one or multiple instances of trained objects in the current camera frame (Figure 4). We extract the image features with the well-known *scale invariant feature transform* (SIFT) [16], or use the so-called *speeded up robust features* (SURF) [7]. While RFID tags uniquely identify a product, the vision-based approach can only assign the product class to a package (for example “Lemon Muffins”). Additionally, in contrast to product identification using RFID, the vision-based technique requires a free line-of-sight between the camera and the product. On the other hand, materials like metal or water in the vicinity of tagged products influence the detection rate of RFID, but not the detection rate of the vision-based method. Unlike RFID, the visual object recognition provides the bearing of the product relative to the robot. This bearing can be computed from the position of the product in the image. Thus, RFID and vision complement each other. We plan to fuse both product detection approaches to improve the robustness and the detection rate.

Figure 4 Vision-based product recognition. The package shown in the upper part of the figure is contained in the database. The lower part of the figure shows the current view of the robot, in which the product is detected twice.

2.3 World Model

The world model acts as the central point of integration for the connected subsystems. Communication with the connected mobile agents is performed using inter-process communication provided by the CARMEN toolkit [2] to fulfil real-time requirements. The ERP and visualization systems are connected with the world model following the SOA paradigm using web service interfaces and the SOAP-over-HTTP protocol.

Another key feature of the world model is the provision of different maps of the environment. These maps are based on different types of sensory information acquired by RFID readers, Bluetooth nodes, WLAN access points, and laser scanners. The connected agents use the maps for self-localization and path planning.

2.4 ERP/Database

In order to enrich the data gathered by robots and other mobile agents with detailed product specific data, a connection from the AmbiSense system to a company's ERP system is needed. Typically the ERP system is the place where companies store all product related data concerning the production and warehousing of produced goods. This data store is not the only place housing product related data in a company, though. In most cases there is also a content management system (CMS) in which the more customer-oriented description and product data are stored. In order to provide the AmbiSense system with a complete set of product information, these two types of systems need to be integrated.

There are two main reasons for not connecting the AmbiSense system directly to the ERP system. First, a direct connection to one specific ERP system would cause a lot of effort if a company were to adopt the AmbiSense system into their existing application landscape. Second, unlike a stationary RFID system, RFID systems attached to a mobile agent like a robot cause a much larger data flow that cannot be directly linked to a value creation process. If the number of mobile clients is scaled up significantly, the connected ERP system would be exposed to a large number of requests, which would drastically slow down the system. In order to regain normal performance levels, the ERP system would have to be scaled up as well. This turns out to not be cost-effective. The reasons stated above lead to an architecture which abstracts from the ERP system and focuses on the integration using a so-called ERP integration server. The SOA paradigm provides the benefits needed and was therefore used as the archetype for the architecture implemented. It is realized through web services, which provide an interface described by the web service description language (WSDL, [9]). In this context, the data types used by the different applications was identified and specified using the XML schema definition language (XSD, [10]). The web services are aggregated on an application server running a business process execution language (BPEL, [6]) process engine. This allows for easily reusing the designed business processes in other complex scenarios. Figure 5 shows the technical architecture of the ERP integration server and the con-

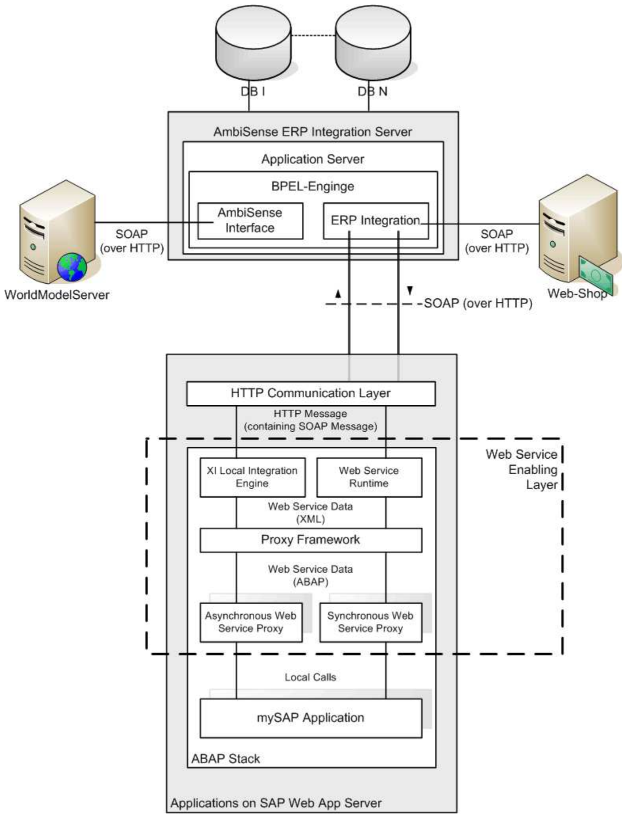

Figure 5 Technical architecture of the "ERP integration server".

nected systems. The ERP system depicted in this example is mySAP2005. Technically it would be possible to host the ERP integration server on any application server of the application landscape. In this case it would even be possible to host it directly on the application server of mySAP2005 (NetWeaver2004s). This approach was not taken, as one of our goals is vendor neutrality.

The main functions of the ERP integration server are the caching of product information and the provisioning of the flexible means of integration of the backend systems. The focus is on the business process implemented in the application scenario while offering stable interfaces to the connected AmbiSense systems. Another important factor favoring this decision is the ability to integrate other standard applications. Examples of applications integrated into the AmbiSense system landscape are a tag data translation (TDT, [20]) service, an electronic product code information service (EPCIS, [21]) and the application level events (ALE, [22]) service. These services all follow the GS1/EPCglobal [3] standards. The designed business processes using the mentioned services are kept in the form of BPEL choreographies. BPEL allows the composition of services into one choreography, which then is offered as a single web service to external applications. A major advantage of this feature is the reuse of business processes in other more complex business processes. BPEL allows for a complete reorganization of a business flow without changing the overall interface of the business process if the starting and final activity are left unchanged. This provides great flexibility with respect to the integrated services that are used in the modeled business processes. These layered processes offer stable web service interfaces to the world model server. The communication between the systems connected to the ERP integration server utilizes the SOAP-over-HTTP standard.

2.5 Visualization and Interaction

This part of the system aims at creating a graphical human-machine interface (HMI) for the complete system of autonomous mobile systems and ambient sensors. The HMI allows the user to extract relevant data from stationary and mobile sensors like RFID readers. A 3D model of the surrounding is used to depict the spatial reference of information and therefore is the key instrument for intermediation of data and human being. The visual output can be displayed ubiquitously: not only at the control room but also on mobile devices and on web pages via the internet. This way, information of both static and dynamic sensors like RFID tags and associated data can be placed in the view at appropriate positions.

According to our application scenario different types of information have to be handled by the visualization system: Localization data of robots and RFID tags are shown in the virtual scene and additional product information, e.g. ERP data from the backend database, can be retrieved and displayed.

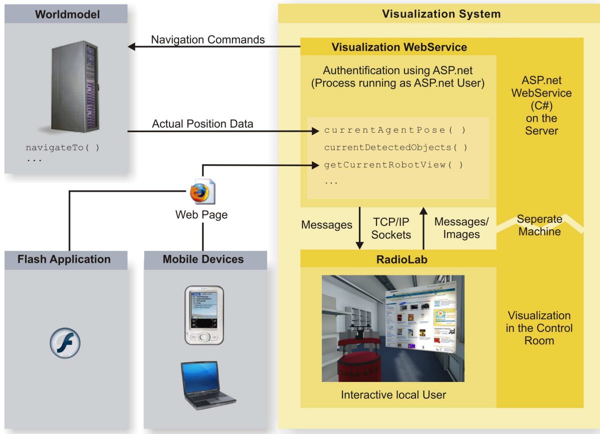

Figure 6 Architecture of our visualization web service and connection to *RadioLab*.

A HMI was developed based on our warehousing and retail scenario by applying principles for intuitive and efficient navigation and visualization [25]. The HMI satisfies different criteria that are motivated by the fact that potentially many mobile agents (robots, staff using PDAs, etc.) are in use and these have to be coordinated. Many different multimodal types of information have to be processed and the spatial reference of the data must be accessible to the user. Depending on the user, different types of information are relevant and these data are presented in a task-related way. Since the users generally use different hardware (laptops, PDAs, mobile phones, etc.), this demands for a visualization that can be displayed ubiquitously. The three-dimensional representation of the environment alleviates the coordination of the mobile agents and points out the spatial relationship of the data in an intuitive way. The visualization itself is not limited to a classical 3D view, but the three-dimensional data allows for the generation of various task-related visualizations. Furthermore, these can be specifically tailored to the needs and characteristics of different visualization hardware (e.g. small display, low performance, low bandwidth for communication, etc.). Some examples of the different types of visualization using different non-photorealistic rendering (NPR) techniques to achieve a higher level of legibility are shown in Figure 7. The rendering technique is similar to the method proposed by Fischer et al. [11], but makes use of the radiosity values stored in the vertices of the refined meshes instead of evaluating a local lighting model. Additionally, silhouettes are determined in object space, then subdivided and displaced perpendicular to the viewing direction and finally blended on top of each other to account for a natural pencil or ink stroke.

The software our visualization system is based on is *RadioLab*, a visualization and presentation module developed at the WSI/GRIS. It features interactive visualization of highly realistic lighting scenarios in real-time. We extended *RadioLab* to connect to our web service for the purpose of

(a) Pencil drawing using material colors. (b) High contrast cell shading.

Figure 7 Examples of different NPR techniques. For a photorealistic rendering see Figure 8(b).

(a) Robot in front of supermarket shelf. (b) Photorealistic real-time rendering.

Figure 8 Comparison of real-world demonstrator and virtual scene.

receiving and sending the types of information mentioned above. A schematic overview is given in Figure 6. The loose coupling to the web service using TCP/IP allows for running the web service and *RadioLab* on seperate machines as well as connecting multiple instances of the visualization to the web service. This way the concept of SOA is consistently carried on. The system provides for easy extensibility, flexibility and reusability of the communication and visualization components.

All mobile agents (robots, etc.) are displayed in the virtual environment at their estimated positions. The user can freely navigate in the scene but also view the scene from the point of view of the robot ("chase-cam" mode). Parallel to this, all identified products are listed according to their detection time.

We solved the task of visualizing the ERP data by incorporating the HTML rendering engine Gecko (used by e.g. the web browser Firefox) into *RadioLab* using a technique similar to [4]. This way product specific data can be displayed at the actual position of the product in the 3D scene. For now, the product homepage is shown, but in the future we will use the web frontend provided by the ERP software to display a dynamically generated web page that enables the user to view and edit the product data in 3D in real-time. The key benefit is a very intuitive interface as well as a loose coupling of functionality and visualization. A screenshot of the virtual scene along with the user interface and product specific information is shown in Figure 8(b).

Summarizing, the visualization supports the user in planning the operation, observating the mobile agents, localizing the products and distributing the information.

In the future, we plan to generate the 3D model of the surrounding in a semi-automatic way using laser scanners, depth cameras and stereo imaging techniques [8, 15]. A detailed three-dimensional representation of the environment serves not only as input data for the visualization itself but will also provide for an easier localization of the mobile agents in case of missing or damaged sensors.

3 Results

We implemented the overall system as described in the preceding section and tested it under near-to-real-world conditions. We presented a live demonstration of our system in September 2007 at the AmbiSense Workshop in Tübingen in presence of an audience composed of representatives from industry and universities (Figure 8(a)). As a first typical tasks for our scenario, we showed how the robot explored the supermarket and found a specifically chosen product. Another feature that we presented to the audience was the control of the robot by selecting the goal position in the virtual scene. In this context, the products in the vicinity of the robot were enriched with additional product data from the backend system. This data was visualized in the form of web sites in the virtual scene and on the touchscreen display of the robot. While performing these tasks, the robot localized itself using RFID, WLAN and Bluetooth. All in all, the AmbiSense system architecture proved itself as a robust and reliable basis for future research and development.

The detection rates directly in front of our RFID antennas (1 m) are approximately 60%, and lower in regions that have a greater distance (Figure 3). Due to these detection rates, the robot typically must pass a shop shelf several times before it has scanned almost all products in the shelf. Note that the sensor model shows average detection probabilities over all orientations of the tags with respect to the antenna. If a tag is oriented parallely to the antenna, the detection probabilities can be significantly larger.

The localization accuracy in our supermarket scenario depends on the types of sensors used. In case of RFID the average localization error was below 0.3 m. When using Bluetooth and a particle filter, the localization error was below one meter in half of the cases. Utilizing the WLAN technique, the average localization error is a few meters. For a detailed discussion see [23].

4 Conclusion and Outlook

We have presented a novel prototype system for robot-assisted inventory. Our developed components revealed a successful interplay in this scenario and the application of principles of SOA for communication proved itself to be very efficient and flexible.

Current research focuses on the following aspects: Firstly, the fusion of different types of sensor data such as RFID, Bluetooth and WLAN in order to further improve self-localization of the mobile agents. To further increase the RFID detection rates, we plan to test different RFID readers and will develop algorithms to exploit the reader models optimally in inventory scenarios.

Through the experiences in our demo scenario, we have identified the access to the backend system as a bottleneck for the provisioning of product data for the connected systems. Therefore, the third aspect is the optimization of the average access time to product data in the backend systems by means of a product cache.

Lastly, we also plan to integrate live video streams in the visualization and develop interfaces for accessing and displaying 3D visualization data on mobile devices.

5 Acknowledgments

This work was funded by the Landesstiftung Baden-Württemberg in the scope of the BW-FIT project AmbiSense.

6 References

[1] AmbiSense Project Homepage. URL: `http://www.ambisense.org`, April 2008.

[2] Carmen robot navigation toolkit. URL: `http://carmen.sourceforge.net`, April 2008.

[3] GS1/EPCglobal inc. URL: `http://www.epcglobalinc.org/standards/`, April 2008.

[4] LLMozLib and uBrowser. URL: `http://ubrowser.com/`, April 2008.

[5] METRO group RFID innovation center. URL: `http://www.future-store.org`, February 2008.

[6] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. *Business Process Execution Language for Web Services version 1.1 (BPEL4WS)*. OASIS, May 2003.

[7] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV 06)*, pages 404–417, Graz, Austria, May 2006.

[8] Peter Biber, Sven Fleck, and Wolfgang Strasser. The wägele: A mobile platform for acquisition of 3d models of indoor and outdoor environments. In *9th Tübingen Perception Conference (TWK 2006)*, 2006.

[9] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. *Web Services Description Language (WSDL) 1.1*. W3C, March 2001.

[10] D. C. Fallside, P. Walmsley, H. S. Thompson, D. Beech, M. Maloney, N. Mendelsohn, P. V. Biron, A. Malhotra, M. Holstege, and A. S. Vedamuthu. *XML Schema Version 1.0*. W3C, October 2004.

[11] J. Fischer, D. Bartz, and W. Straßer. Stylized Augmented Reality for Improved Immersion. In *Proceedings of IEEE Virtual Reality (VR 2005)*, pages 195–202, 2005.

[12] A. Günther and C. Hoene. Measuring round trip times to determine the distance between wlan nodes. In *Proceedings of NETWORKING*, pages 768–779, Waterloo, Canada, 2005.

[13] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and Localization with RFID Technology. In *Proc. IEEE International Conference on Robotics and Automation (ICRA 2004)*, pages 1015–1020, New Orleans, LA, USA, 2004.

[14] C. Hoene. Goodtry, software for WLAN-trilateration. URL: `http://sourceforge.net/projects/goodtry/`. Expected release date: July 2008.

[15] Benjamin Huhle, Sven Fleck, and Andreas Schilling. Integrating 3d time-of-flight camera data and high resolution images for 3dtv applications. In *3DTV CON - The True Vision*, May 7-9 2007.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[17] S. Schneegans, P. Vorst, and A. Zell. Using RFID Snapshots for Mobile Robot Self-Localization. In *Proc. European Conference on Mobile Robots (ECMR 2007)*, pages 241–246, Freiburg, Germany, September 19-21 2007.

[18] S. Subramanian, J. Sommer, S. Schmitt, and W. Rosenstiel. SBIL: Scalable indoor localization and navigation service. In *Proc. Third International Conference on Wireless Communications and Sensor Networks (WCSN 2007)*, 2007.

[19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[20] K. Traub, C. Asher, G. Gibert, R. Swan, and G. Morgan. EPC tag data translation (TDT) standard. http://www.epcglobalinc.org/standards/tdt/, January 2006.

[21] K. Traub, C. Asher, G. Gibert, R. Swan, and G. Morgan. EPCIS - EPC information services standard. http://www.epcglobalinc.org/standards/epcis, September 2007.

[22] K. Traub, M. Frey, R. Bach, D. Paley B. Tracey, and S. Chowdhury. Application level events (ALE) standard. http://www.epcglobalinc.org/standards/ale, Feburary 2008.

[23] P. Vorst, J. Sommer, C. Hoene, P. Schneider,

C. Weiss, T. Schairer, W. Rosenstiel, A. Zell, and G. Carle. Indoor positioning via three different rf technologies. In *Proc. 4th European Workshop on RFID Systems and Technologies (RFID SysTech 2008)*, Freiburg, Germany, June 10-11 2008.

[24] P. Vorst and A. Zell. Semi-autonomous learning of an RFID sensor model for mobile robot self-localization. In *Proc. European Robotics Symposium (EUROS 2008)*, pages 273–282, Prague, Czech Republic, 2008.

[25] B. Zhu and H. Chen. Information visualization. *Annual Review of Information Science and Technology*, 39(1):139–177, 2005.

Timo Schairer studied computer sciences at the University of Tübingen, Germany, and received his master's degree (Diplom) in 2007. Since then, he has been a Ph.D. candidate with University of Tübingen, at the Department of Graphical-Interactive Systems led by Prof. Wolfgang Straßer. His research focuses on interactive real-time visualization.
Contact: schairer@gris.uni-tuebingen.de

Christian Weiss received the diploma in computer science from the University of Tübingen, Germany, in 2005. Currently, he is a Ph.D. candidate with University of Tübingen at the Department of Computer Architecture led by Prof. Andreas Zell. His research interests include mobile robotics and computer vision.
Contact: c.weiss@uni-tuebingen.de

Patrick Schneider received the diploma in computer science from the University of Stuttgart in 2006. He is currently a Ph.D. candidate with University of Stuttgart at the Institute for Mechanical Engineering (IAT) at the University of Stuttgart. His research interests lie in the field of service-oriented architectures and integration of RFID systems into Enterprise Ressource Planning (ERP) Systems.
Contact: patrick.schneider@iao.fhg.de