Service-Oriented Integration of Multi-Sensor Environments into ERP Systems

Patrick Schneider
Oliver Höß
Anette Weisbecker

University of Stuttgart Institute for Human Factors and Technology Management Stuttgart, Germany

Fraunhofer Institute for Industrial Engineering (IAO) Stuttgart, Germany

In order to make the vision of the "real time enterprise" a reality, a wide range of sensors, potentially spread across the whole company, must be integrated into the existing IT infrastructure. The benefits of sensor integration are probably most drastically in the field of warehouse- and supply chain–management. Sensors typically used in a real world scenario are for example RFID, acceleration sensors, temperature sensors, movement sensors etc. Currently the typical approach adopted by most companies is to incorporate proprietary software in order to achieve the desired level of automation for each type of sensor. This results in highly heterogeneous application landscapes which are hard to maintain and hard to integrate. If newly adopted business processes span across business boundaries, this fact causes a lot of severe problems. In this paper an approach is presented, that shows how these problems can be avoided by using a standard based flexible architecture to integrate sensor data into an ERP back end system. The central paradigm applied to the design of the integration architecture is the concept of service-oriented architectures which facilitates the integration of heterogeneous systems. The approach presented in this paper was developed in the context of the AmbiSense project which will also be introduced briefly.

Keywords: application integration, multi-sensor environments, service-oriented architecture (SOA), enterprise resource planning (ERP), radio frequency identification (RFID)

1 Introduction

It has always been the desire of the management of any kind of enterprise to be able to view the state of the company's value creation chain by the push of a button. An example would be an accountant executive officer checking how many open transactions are to be processed. For the accounting part of a company this vision of a "real-time enterprise" [Kuhlin & Thielmann 2005] has already been implemented in some sort way in most companies.

A different, but not less important part of company value creation chain is the production side. Most enterprise resource planning (ERP) suites out on the market today provide application components that offer the means for in depth production planning. Before being able to use these applications on a daily basis, the engaged machinery available for production has to be mapped to the software system in use. In order to make the production of goods as efficient as possible, a good warehouse management and supply chain management have to be in place. At present, the issues depicted in this paragraph are all addressed by pieces of software in most ERP systems. The problem is that there is a mismatch of the state of the software and real life. An example for this mismatch could be the state of a warehouse. As there are humans working in a warehouse, you can be sure that an inventory would reveal a mismatch between the nominal and the actual value.

A means to encounter this problem would be the use of modern sensor equipment directly connected to the ERP back end systems and carefully integrated into existing business processes. A technology that is becoming more and more popular in this area is Radio Frequency Identification (RFID [Finkenzeller 2002]). This technology enables enterprises to track and trace their goods and operations more easily than with bar codes for instance. But RFID is not the only type of sensor information relevant for integration into ERP back end systems. Location or positioning information collected through sensors already in use in companies, like WLAN (IEEE 802.11) or Bluetooth [Bluetooth 2007] present options that could bring significant value increase in some business processes. Integrating sensor information into existing ERP back end systems is a very complex task for many different reasons depicted later.

This article is based on the work of the AmbiSense project, which for the sake of better understanding is introduced in section 2. After this introduction we will delineate the state of technology in section 3 before describing the integration architecture in detail in section 4. We will conclude this paper with a short summary and the future prospects in this area in section 5.

2 Overview of the AmbiSense Research Project

One of the main objectives of the AmbiSense project [AmbiSense 2007], which is funded by the Ministry for Science and Research Baden-Württemberg, is to provide a framework which enables the usage and fusion of sensor data in enterprise reporting software. The focus on the sensors in this research project is RFID, wireless LAN (IEEE 802.11) and Bluetooth. This framework combined with a powerful graphical end user interface in the form of an ambient intelligence display could present significant advantages over present day solutions. Application areas of the AmbiSense project are warehouse and inventory scenarios, logistic scenarios and surveillance tasks.

The project is split into 5 project groups working closely together to achieve a maximum outcome. For the sake of clarity the overall project apportioning is depicted in figure 1. The first part of the project deals with the communication infrastructure both as infrastructure for communication between the application systems and as a means of locating agents based on methods as described in [Günther & Hoene 2005]. The second part of the project deals with autonomic systems and bio-analogue algorithms. Its goal is to facilitate the infrastructure needed to allow for an autonomic system, be it robotic or human, to communicate with the whole system, as well as to gather information from sensors connected to the autonomic agent. The third project is centred on bluetooth technology, used for communication and allocation purposes. In order to allow reasonable interaction by users with the system, the fourth project deals with the presentation of an ambient intelligence user interface. This article is written with a focus on the fifth project, which deals with the integration of the multi sensor system into the ERP back end system.

3 State of Technology

This part of the article describes the state of technology that is relevant both for the sensor side and for the integration altogether. As the main focus of this article lies on the architecture of the whole system and the integration of the ERP back end systems, the sensor side is only covered to the extent that is relevant for the overall architecture. In section 3.1 the relevant technologies for the sensors used in the project are described. Section 3.2 describes the standards used to integrate the ERP back end systems into the project.

3.1 Sensor Related Aspects

The main focus of the AmbiSense project with respect to sensors is delimited to the following types of sensors:

- Wireless LAN (access points and mobile WLAN cards) used for communication and localization
- Bluetooth [Bluetooth 2007] (BTNodes [Beutel et al. 2003]) used for communication and localization
- RFID [Finkenzeller 2002] used for localization and identification
- Laser scanners (robot) used for localization
- Ultrasound (robot) used for localization
- Cameras used for identification

As most of the sensors listed above do not provide direct input for the ERP system, nor require data from the back end systems, the main focus on the sensor side lies on the RFID tags. In this respect there are some standards that are vital for a correct integration. These standards are listed below and briefly described.

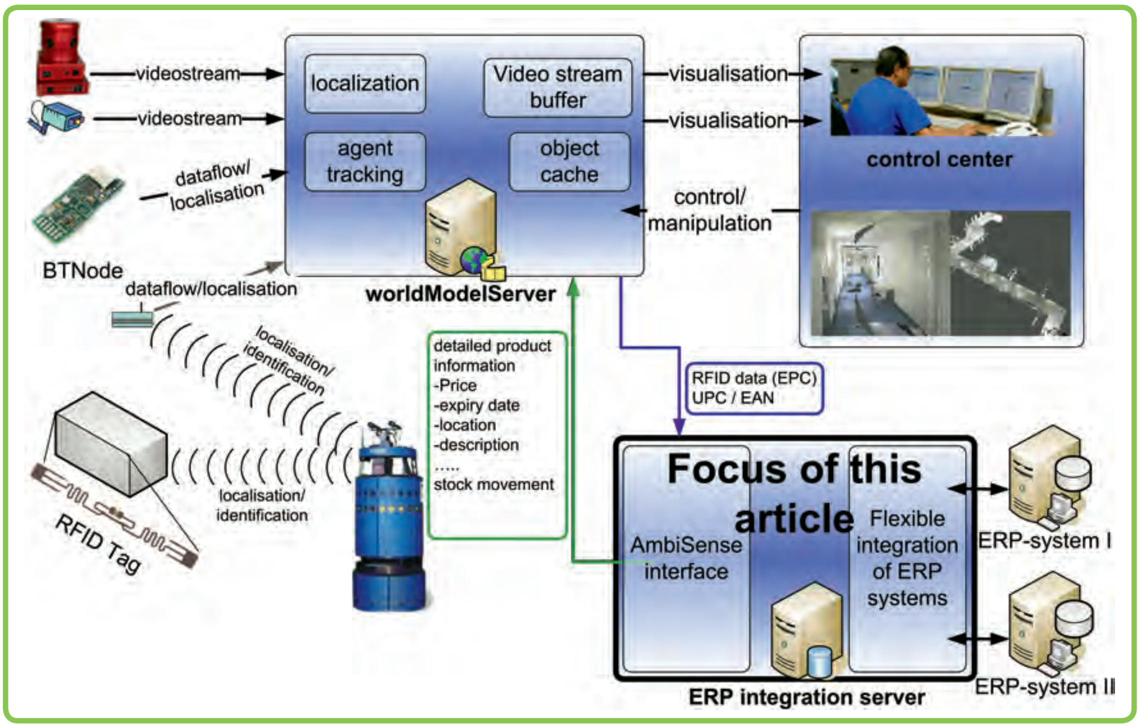

Figure 1: Overview of the AmbiSense Project

- EPC [Asher et al. (a)] – the electronic product code
- EPCIS [Asher et al. 2007 (b)] – the electronic product code information system
- ALE [Asher et al. (c)] – application level events

All of the standards listed above have been adopted by EPCglobal, a subsidiary of GS1, which itself was formed when EAN International, the Uniform Code Council (UCC) and the Electronic Commerce Council of Canada (ECCC) merged. The codes previously provided by these organizations already play a vital role in today's global economy. The EAN Code for instance is used on 90 % of all consumer products. Another widely used code is the UPC code published by the former UCC. Therefore in context of ERP integration with the intent of providing product data, these standards can not be disregarded.

The EPC is a standard that comprises the capabilities of a bar code with additional information such as serial number and individual information on an RFID chip that allows contact-free and bulk reading [Asher et al. (a)].

The EPCIS standard describes an information system that stores data about collected EPC information in the vicinity of a company. This could for instance be the results of bulk readings at an RFID reader located at a warehouse entry door [Asher et al. 2007 (b)].

Application Level Events (ALE) presents a standard that is used to filter the data collected by RFID readers and stored in an EPCIS. ALE provides a framework that makes it possible for an application to subscribe to specialized events and filter data, so that it only receives data relevant for the application [Asher et al. (c)].

3.2 Architectural Aspects

The key concepts behind the architecture of the integration layer of the AmbiSense project, which is the main focus of this article, are amongst others derived from the following requirements:

- Flexible adoption of different kinds of ERP back end systems
- Implementation neutrality
- Easy extensibility
- Vendor neutrality.

A simple yet very powerful approach that fulfils the requirements mentioned above is the paradigm of service-oriented architectures (SOA) [Weerawarana et al. 2005]. Based upon open common internet standards, services are implementation-neutrally described via the Web Service Description Language (WSDL) [Christensen et al. 2001]. Almost any proprietary application can easily be fitted with a web service interface and embedded into a business process using the Business Process Execution Language for Web Services (BPEL4WS, short BPEL) [Andrews et al. 2003]. A business process described through BPEL can itself be presented as a web service and embedded further into a new process. These facts make SOA the first choice in this project.

4 The Integration Architecture

Keeping the requirements and standards described in section 3 in mind, the focus of this part of the article lies more on the actual design instead of the premises presented above. Service-orientation allows for the focus to be set more on the process part of the integration rather than on the technical layer. For a better understanding figure 2 shows a coarse overview over the archi-

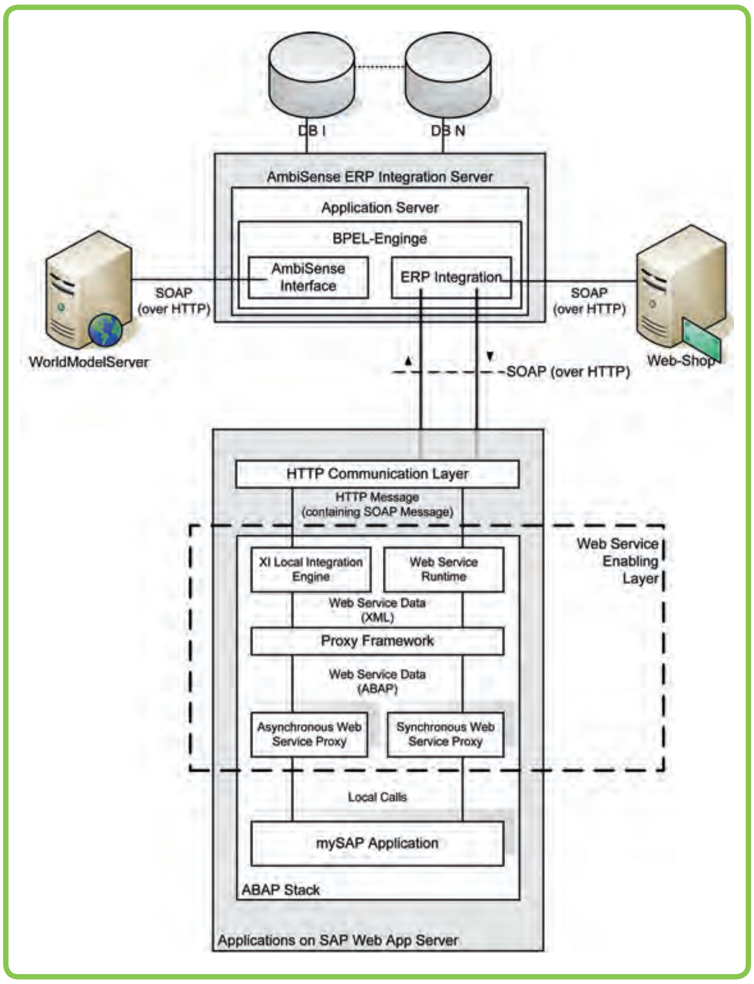

Figure 2: AmbiSense Overall Architecture

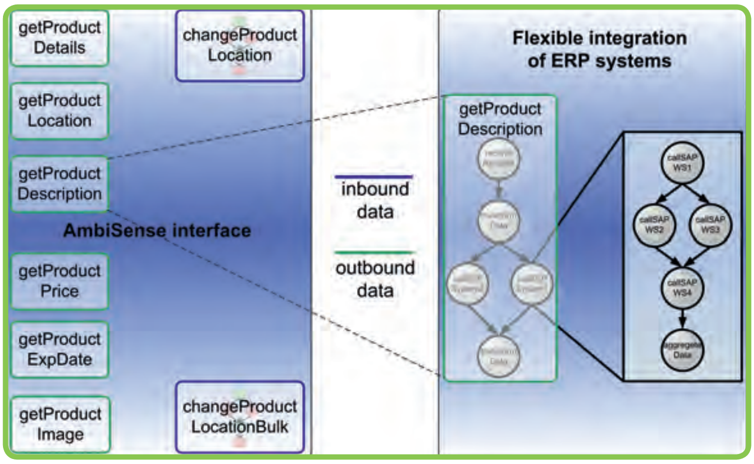

Figure 3: Architectural Concept of the AmbiSense ERP Integration Server

tecture used in the AmbiSense project.

A "world model server" is used for the storage of different kinds of maps used for localization of the agents and objects in the "hemisphere" of the server. In the first version of the server, the project team has decided to store five different types of maps on the server.

- The first kind of map is a floor plan, which is acquired using a laser scanner. It presents the basis for the localization of the agents. Robots or agents can utilize this map to navigate through the area covered by the world model server.
- The second kind of map is a WLAN map into which the stationary WLAN access points are inserted. This is also used for localization purposes [Günther & Hoene 2005].
- Another kind of map used and stored on the world model server is the object map. Objects are identified using RFID equipment consisting of a reader and the tag used to identify the location or object.
- Analogue to the WLAN map the fourth map is the bluetooth map in which all active BTNodes are listed. There can be stationary BTNodes and BTNodes attached to agents.
- The fifth and final map type is the map used by the graphical user interface. It presents a 3-dimensional space generated out of the data collected by the "Waegele", a mobile platform for the acquisition of 3D models [Biber et al. 2006].

The communication of the agents with the ERP integration server will always be served through the "intermediary" world model server. This is done for two reasons. The first being the redundant interfaces that would have to be provided if the agent would be allowed to communicate with both the integration server and the world model. The second and cogent reason for this intermediary approach is the vast amount of traffic that could be induced on the ERP system if a rising number of agents would be involved in a real world scenario. If all requests by all agents were always serviced by the ERP integration server, the connected ERP systems would have to be scaled up significantly in order to be able to service the rising amount of requests imposed upon them. In this context the world model server can be seen as an object cache similar to the EPCIS. In order to fulfil some of the communication requirements by the ERP integration server, the world model server also comprises an ALE service as mentioned in section 3.1. This is used when an event needs to be propagated to the control centre or the ERP integration server and vice versa. In order to make the architecture more figurative a standard usage scenario is shortly described in the following few lines.

If an agent identifies an object that has not been identified before, the object data (EPC) is stored in the world model server. If the control centre or an agent requires more information about an object, the intermediary (world model server) queries the ERP integration server through a Web Service call and relays the results back to the agent. In order to keep the workload balance of the connected ERP systems as low as possible, the world model server caches the object information passed to the requesting agent. If another agent queries the same object again, the results are served by the world model server's cache. In order to keep the cache of the world model server up to date, the integration server communicates any changes made to objects/products inside of the connected ERP systems to the world model server.

The ERP integration server is based on an application server that supports BPEL processes. The actual integration process of the integration of different ERP systems is thereby reduced to the modelling of the processes necessary for the interaction in the project context on a non-technical business level. As BPEL allows a process to expose itself as a web service, more complex processes can be modelled without over straining the modeller. The interface of a BPEL process is determined by exactly two activities of the process. These activities are the starting (receiving) activities at the beginning of a process and the final (reply/invoke) activity at the end of the process. The overall integration structure of the ERP integration server is outlined in figure 3.

For reasons of understanding figure 4 shows a sample process as it is used on the integration server. It depicts the request for product description from an attached SAP ERP system which is invoked through a Web Service interface exposed directly through the new Enterprise Service Repository (ESR) advocated by SAP. In order to keep the exposed interfaces stable, the start and end activities are not changed throughout the life cycle of the process modelled. To fulfil this requirement, the exchanged data was modelled at the beginning of the project. This had to be done very thoroughly by analysing the business process with respect to the business objects required for the successful execution. This was done by means of a XML schema definition (XSD). To fulfil the differing needs of the integrated ERP systems a data conversion has to be done. In figure 4 this conversion is implemented before and after the activity "invokeSAPClient". The first assign activity converts the relevant parts of AmbiSense specific data structure to the SAP specific structure.

By replacing the "invokeSAPClient" activity along with its corresponding assign activities and replacing them with an invoke activity on another ERP system that supports Web Services like, SAGE, Compiere or the Oracle e-business suite, the interface is not affected as mentioned above. Only the transformation of the data types needs to be changed.

5 Summary and Future Prospects

In this paper we described the overall architecture of the AmbiSense system. It uses two central servers to integrate the attached agents and ERP systems. The focus of this paper was the architecture of the ERP integration server which integrates different ERP systems by using web service technologies and a flexible process orchestration and transformation layer. The integration of sensory data like RFID or motion sensor data can improve business processes, for example in the supply chain, drastically. Use Cases for efficient RFID utilization for instance can be found at [ECC Stuttgart-Heilbronn].

Based on a service-oriented architecture, the platform is set for future extensions and integration into larger scaled architectures. By using scalable technologies like application servers and business process engines the system itself can be scaled to service a large amount of clients connected.

6 Acknowledgements

The authors would like to thank the Ministry for Science and Research Baden-Württemberg for funding the project AmbiSense.

7 References

AmbiSense homepage, http://www.ambisense.org, last visited 16.4.2007

Andrews, T.; Curbera, F.; Dholakia, H.; Goland, Y.; Klein, J.; Leymann, F.; Liu, K.; Roller, D.; Smith, D.; Thatte, S.: Business Process Execution Language for Web Services version 1.1 (BPEL4WS). OASIS, http://www-128.ibm.com/developerworks/library/specification/ws-bpel/, May 2003

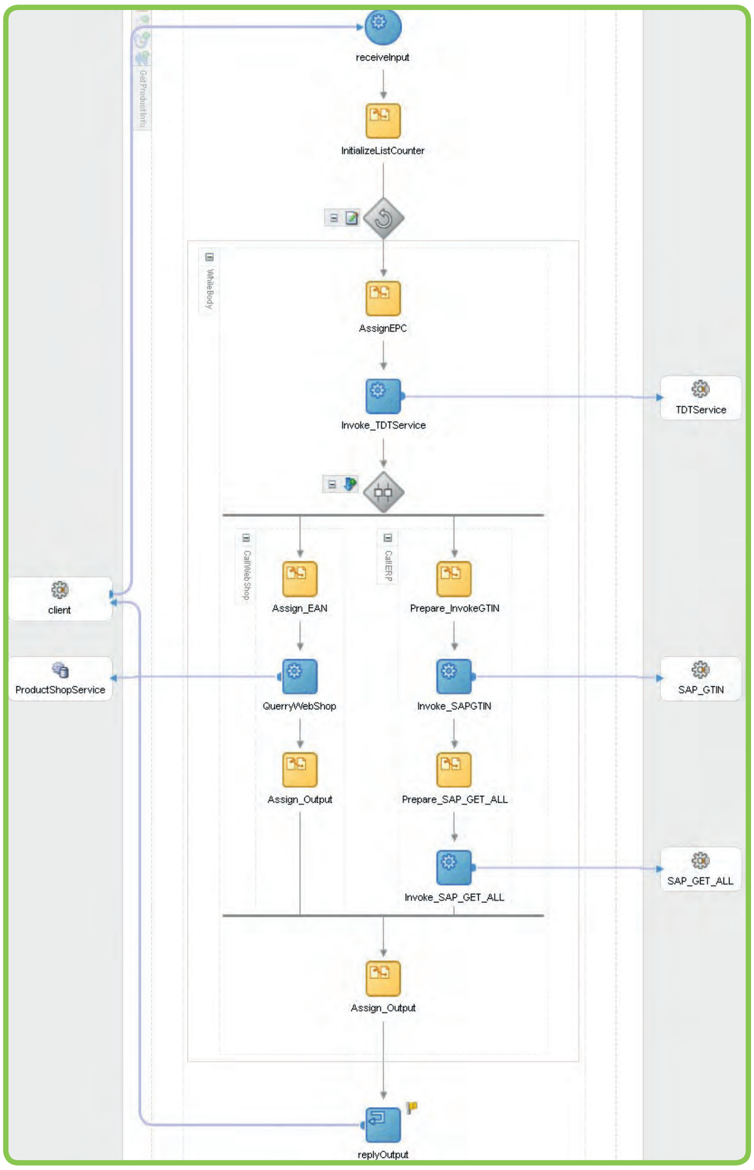

Figure 4: Screen Shot of a Sample Integration Process

Asher, C.; Gibert, G.; Swan, R.; Traub, K.; Morgan, G. (a): EPC Generation 1 Tag Data Standards Version 1.1 http://www.epcglobalinc.org/standards/EPCglobal_Tag_Data_Standard_TDS_Version_1.1_Revision_1.27.pdf

Asher, C.; Gibert, G.; Swan, R.; Traub, K.; Morgan, G. (b): EPC Information Services Version 1.0 http://www.epcglobalinc.org/standards/EPCglobal_EPCIS_Ratified_Standard_12April_2007_V1.0.pdf

Asher, C.; Gibert, G.; Swan, R.; Traub, K.; Morgan, G. (c): The Application Level Events (ALE) Specification, Version 1.0, http://www.epcglobalinc.org/standards/Application_Level_Event_ALE_Standard_Version_1.0.pdf

Beutel, J.; Kasten, O.; Mattern, F.; Römer, K.; Siegemund, F.; Thiele, L.: Prototyping Wireless Sensor Network Applications with BTNodes, Computer Engineering and Networks Laboratory, ETH Zürich, 2003

Biber, P.; Fleck, S.; Strasser, W.: The Wägele: A Mobile Platform for Acquisition of 3D Models of Indoor Outdoor Environments, in: 9th Tübingen Perception Conference (TWK 2006), 2006

Bluetooth Core Specification v2.0 homepage: http://www.bluetooth.com/Bluetooth/Learn/Technology/Specifications/, last visited 16.4.2007

Christensen, E.; Curbera, F.; Meredith, G.; Weerawarana, S.: Web Services Description Language (WSDL) 1.1, W3C, http://www.w3.org/TR/wsdl, 2001

ECC Stuttgart-Heilbronn: RFID-Atlas, http://www.rfidatlas.de

Finkenzeller, K.: RFID-Handbuch, Hanser Fachbuchverlag; Auflage 3, 2002

Günther, A.; Hoene, C.: Measuring round trip times to determine the distance between WLAN nodes. In: Networking 2005, Waterloo, Canada, May 2005

Kuhlin, B.; Thielmann, H.: The Real Time Enterprise – Facts, practices and prospect, Springer, 2005

Weerawarana, S.; Curbera, F.; Leymann, F.; Storey, T.; Ferguson, D.: Web Services Platform Architecture. Prentice Hall, 2005

Dipl.-Inf Patrick Schneider
Patrick.Schneider@iao.fraunhofer.de

Patrick Schneider has been working at the Institute for Human Factors and Technology Management of the University of Stuttgart (IAT) since 2006 and focuses on the research areas of Enterprise Reporting Platforms, SOA and mobile applications. Currently he is active in the AmbiSense project.

Dipl.-Inf. Oliver Höß
Oliver.Hoess@iao.fraunhofer.de

Oliver Höß works at the Fraunhofer Institute for Industrial Engineering (IAO) since 1999 and heads the business unit Software Technology (www.swm.iao.fhg.de) as well as the m-Lab of the Fraunhofer IAO (www.m-lab.iao.fhg.de). He is active in a large number of projects in the areas of software engineering, software architecture and mobile applications.

Priv.-Doz. Dr.-Ing. habil. Anette Weisbecker
anette.weisbecker@iao.fraunhofer.de

Dr.-Ing. Anette Weisbecker is Assistant Director of the Fraunhofer Institute for Industrial Engineering IAO and heads the Competence Center Software Management, which supports companies by the design and implementation of complex software and services. Her focus of research is Software Management, Electronic Business and Grid Computing.