

Towards a Reference Architecture for an Integration Platform for Diverse Smart Object Technologies

Sebastian Lempert, Alexander Pflaum

Center for Intelligent Objects ZIO
Fraunhofer Institute for Integrated Circuits IIS
Dr.-Mack-Str. 81
90762 Fuerth, Germany
{sebastian.lempert | alexander.pflaum}@iis.fraunhofer.de

Abstract: Although the cost-effective integration of diverse smart object technologies like radio-frequency identification (RFID), real-time locating systems (RTLS), and wireless sensor networks (WSN) into existing enterprise infrastructures is very important for companies, integration aspects are ignored by researches frequently. Furthermore it could be observed, that existing commercial middleware products aiming at integration ignore the trend of consolidation between diverse smart object technologies by focusing on passive RFID only. This paper introduces the concept of an integration platform for diverse smart object technologies together with an abstract software architecture. The functionalities that such a platform should provide are derived from the requirements of real world applications and aim at drastically reduced integration costs.

1 Introduction and motivation

Since managers have to plan, organize, staff, direct, and control an organization or task in order to accomplish desired objectives efficiently [Ma69] additional information that supports the management process is very valuable. The statements “you can't control what you can't measure” [De82] and “if you can't measure it, you can't manage it” [Ga93] describe a simple but fundamental insight according the management process that was made early in the fields of computer science and business administration, respectively. Nowadays smart object technologies like RFID [Fi08], RTLS [Ma09], and WSN [KW07] are used more and more to gather information from their environment previously not available and are thus the prerequisite for efficient business processes.

Objects from the real world like containers, palettes and assets in general are called intelligent or smart objects if they are tagged with smart object technologies. Smart objects come from a number of different technology areas and scientific disciplines including embedded systems, ubiquitous and pervasive computing, mobile telephony, telemetry, wireless sensor networks, mobile computing, and computer networking with each area making its own imprint on the technology [VD10].

In research there is consensus that ideal smart objects have unique identifiers and are able to save and process information, to monitor their environment with sensors, to interact with their environment with actors, and to communicate with their environment wirelessly [Sá09], [VD10]. From the authors point of view ideal smart objects are also able to detect their own position in the two- or three-dimensional space with the help of additional infrastructure.

By looking at ongoing standardization processes at EPCglobal [Di09] or DASH7 alliance [Ri10] concerning active RFID it could be observed that in future active RFID tags will have their own power supply, additional sensors, and the ability to communicate with other tags without using RFID interrogators. Another indicator for the increasing consolidation between diverse smart object technologies is the fact that positioning is not a unique feature of highly specialized RTLS anymore and can be done with RFID [Mo09] and WSN [KW07] too.

Since the aim of enterprises is profit they will introduce and integrate smart object technologies only if benefits are higher than costs. Unfortunately the integration of diverse smart object technologies into existing enterprise systems is a complex, time-consuming, and expensive task: integration costs can add up to “35 to 50 percent of all systems development” [Br06] and are thus a significant cost driver. As a result enterprises will use smart object technologies only if integration can be done with reasonable efforts. Despite this fact academic research on wireless sensor networks frequently ignores integration aspects [Ra08]. In contrast in the field of RFID commercial middleware products are established that aim at efficient integration [Le04]. Unfortunately these products ignore the trend of consolidation between diverse smart object technologies by focusing on passive RFID only. Thus the consolidation of diverse smart object technologies demands new integration platforms.

We believe that there is a need for an integration platform for diverse smart object technologies. Therefore in this paper the following research questions are examined:

1. What is an integration platform for diverse smart object technologies and how does this term relate to the term middleware?
2. How should such an integration platform be designed and what functionalities are needed in order to be able to support heterogeneous applications of smart object technologies efficiently?

The rest of the paper is organized as follows. In section 2 we line out the differences between the terms middleware and integration platform and derive a first definition for the term integration platform for diverse smart object technologies. In section 3 we introduce an abstract software architecture for an integration platform for diverse smart object technologies and line out, what functionalities such a platform should provide and what standards should be considered. Furthermore we compare the proposed functionalities with the requirements of different real world applications. In section 4 we compare our concept with related work. We conclude this paper with a summary and an outlook on prospective future work in section 5.

2 Differentiation of terms

Middleware: To the best of our knowledge the term middleware was first used and coined in order to describe software that sits between applications and an operating system in 1968 [NR68]. Unfortunately since then the term was used in many different ways with different meanings in different contexts in literature. Therefore it is very hard to give a clear definition of the term middleware that is generally applicable. Nevertheless from the authors point of view the definition “Middleware is any software that allows other software to interact” [De10] seems to be adequate.

Integration platform: An integration platform denotes a middleware product or a combination of middleware products that aims at connecting different enterprise applications and supports the concept of enterprise application integration [LLS06], where “EAI is the unrestricted sharing of data and business processes among any connected applications and data sources in the enterprise” [Li99].

Integration platform for diverse smart object technologies: An integration platform for diverse smart object technologies like WSN, RFID, and RTLS is an integration platform, which unites these technologies with a shared technology abstraction layer, controls the interaction between these technologies and existing enterprise infrastructures, supports intra-corporate and cross-company integration, and aims at reducing integration costs significantly. Furthermore from the authors’ point of view such an integration platform should be designed as proposed in section 3.

3 Software architecture

The abstract software architecture introduced in this section is intended to act as a template or blueprint for designing integration platforms in the domain of diverse smart object technologies. This intention comes close to the one that is often attributed to reference architectures: “A reference architecture is a generic architecture that can be applied in several types of industries but that share one or more common domains. Reference architectures should provide architectural guidance, best practice information, and should act as a blueprint for designing information systems.” [AP03]. In addition a “reference architecture need not be a concrete architecture; i.e., depending on the requirements being addressed by the reference architecture, it may not be necessary to completely specify all the technologies, components and their relationships in sufficient detail to enable direct implementation” [OA09]. Nevertheless as with reference models, that could be interpreted as the basis for describing reference architectures [OA06], an independent third party should decide whether certain architectures are accepted as reference architectures by applying the architecture in question successfully at least once [Th06]. Thus the authors decided not to declare their proposal as a reference architecture.

In order to have a flexible architecture for an integration platform for diverse smart object technologies that supports a wide range of smart object applications and a wide range of existing enterprise information systems the architecture should be constructed according to the service-oriented architecture (SOA) design paradigm. Since an enterprise service bus (ESB) could be defined as an up-to-date architectural style paradigm to implement a SOA we propose an abstract software architecture that is build upon an ESB [Fu09], [Th09]: as can be seen in Fig. 1 an integration platform for diverse smart object technologies should consist of several modules or services that are interconnected via an ESB. In the following we will explain the functionalities of the different modules/services that have been derived from the requirements of real world applications in more detail.

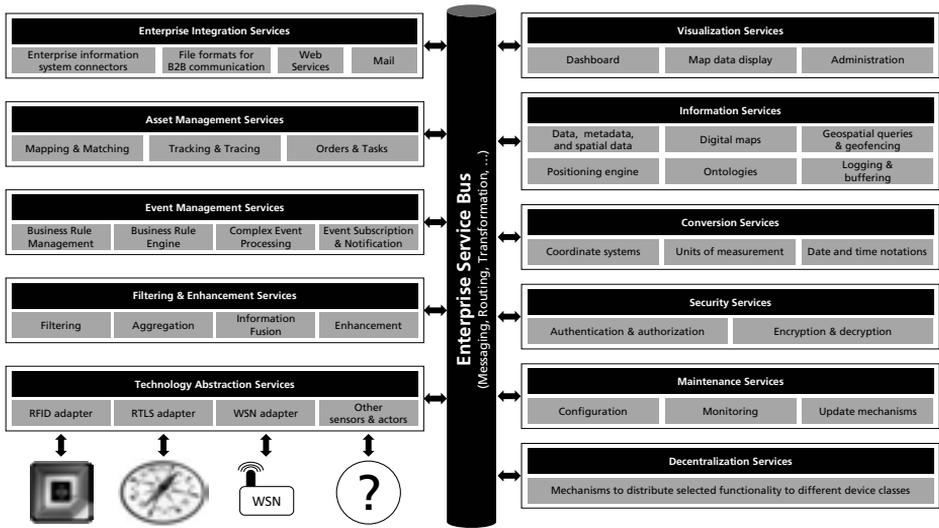


Fig. 1: Software architecture for an integration platform for smart object technologies

Technology abstraction services: Diverse smart object technologies should be accessed from other modules of the platform via a common technology abstraction layer. In detail the technology abstraction layer should enable the communication with and the administration of arbitrary smart object devices in a standardized way while ensuring that the functionality of concrete devices of a particular category can survive. For example simply creating the illusion that a sensor node is nothing more than a passive RFID tag is insufficient since a huge part of the functionality of sensor networks would remain unused. Technology abstraction is achieved with the following idea: information gathered by smart object devices is passed to other modules of the platform in a standardized way by transforming this information in so called basic events (see Fig. 2). Since sensor measurements are worthless if you don't know when and where the measurements took place a basic event consists of a smart object device identifier, a timestamp, and a geo-coordinate at minimum. Depending on the smart object device being used a basic event may contain one or more sensor values, corresponding units of measurement, and status information (e.g. battery status) in addition.

Since timestamps, coordinates and units of measurements are not always provided by the smart object technology being abstracted the integration platform has to add missing information approximately. Since timestamps, coordinates and sensor values may be exact or approximated information corresponding quality indicators are added in the technology abstraction module.

Filtering & enhancement services: Smart object technologies produce a large amount of data that needs to be processed, delivered and assessed according to different objectives of different platform modules. Unfortunately not all data is relevant for every module. Thus the integration platform provides mechanisms that allow filtering and aggregation of information depending on the interest of certain modules of the platform. In order to achieve this task the content based router that is part of the ESB is used. As a general rule information gathered from smart object technologies needs to be further enhanced before business rules can be applied. Thus the filtering & enhancement module passes only so called enhanced basic events to other modules of the platform (see Fig. 2): besides all information contained in a basic event an enhanced basic event contains information on the asset that is tagged with the given smart object device and additional information on the location that a given geo-coordinate belongs to (e.g. a certain storage area on the enterprise premises). This additional information is retrieved from a spatial database and/or an enterprise information system that is connected to the platform. Additional enhancements might be achieved with information fusion: “Information fusion denotes the process of combining data from different sensors or information sources to obtain new or more precise knowledge on physical quantities, events, or situations.” [RP07]. On the other hand in case that WSN are used information fusion might already take place within the sensor network itself [NLF07]. Thus the integration platform provides interfaces for information fusion algorithms to be plugged in. Furthermore quality indicators are updated in the filtering & enhancement services if information fusion did take place.

Event management services: The task of the event management is to monitor business processes in real-time in order to create warnings and error messages prematurely if events are recognized that are critical or relevant for the business. The information needed to detect business events is delivered through the technology abstraction module and/or the filtering & enhancement module as explained before. In detail the event management is the module that takes basic events and/or enhanced basic events as input and produces business events as output (see Fig. 2). The event management mechanisms are implemented efficiently by using a business rule management system (BRMS) that comes with a business rule engine (BRE) and a complex event processing (CEP) solution complementary. A detailed comparison of different BRMS and CEP solutions can be found in [Ho09] and [Gu09], respectively. In case that a business event is detected, platform modules and applications that are external to the platform and interested in this type of business event are notified via a publish-subscribe-mechanism that conforms to the standard WS-Eventing.

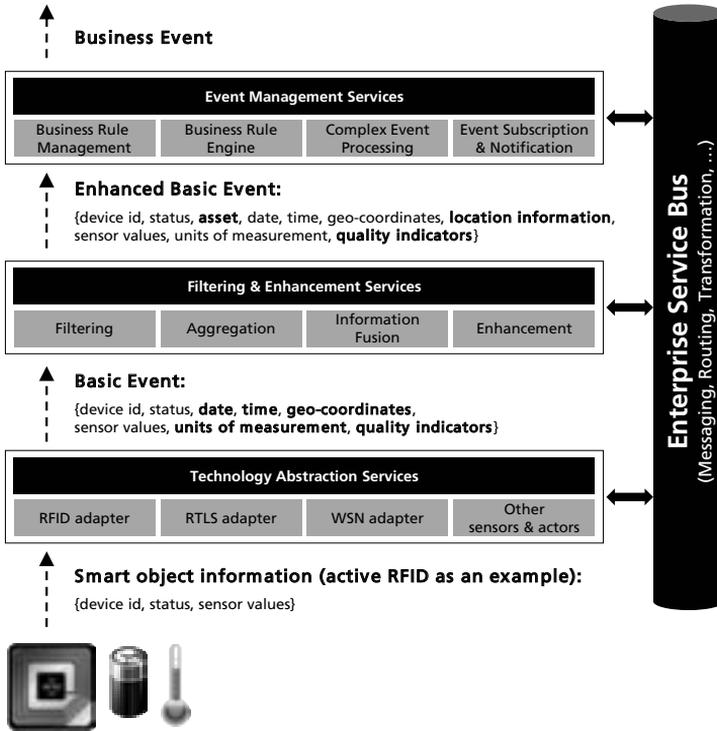


Fig. 2: Transformation of smart object information into business events

Asset management services: As a general rule smart object technologies are used to monitor real world objects. In this regard especially asset management is a popular application of smart object technologies. Thus the integration platform provides mechanisms that allow the efficient implementation of an asset management system: on the one hand the platform retrieves information regarding a mapping between a smart object device and an asset from its own database. In addition the platform retrieves information from its own database and/or ontology indicating whether two or more assets could be matched successfully. For example with this information it is possible to verify whether different hazardous materials could be stored together and whether an item to be placed on a certain palette belongs to the same commissioning order. On the other hand tracking and tracing of assets is very important in practice: while tracking denotes the current location of an asset, tracing denotes the exact sequence of all locations that a certain asset has been at in the past. Since all basic events that are created in the technology abstraction module come with geo-coordinates, tracking and tracing can be easily implemented by simply storing all basic events into the database. Finally the platform provides the ability to retrieve one or more orders that one or more assets corresponds to from its own database or a connected enterprise information system. For example in practice it is necessary to support certain order types like a commissioning order, a production order, a transportation order, etc. In addition it is possible to retrieve a task that a smart object device should carry out on an asset from the database. An example might be the mission to monitor the temperature of perishable goods.

Enterprise integration services: Existing enterprise infrastructures include among other components so called information systems like supply chain management (SCM) solutions, enterprise resource planning (ERP) solutions, warehouse management systems (WMS), customer relationship management (CRM) solutions, and manufacturing execution systems (MES). Since this is a wide palette of software solutions, each with specific interfaces, protocols, and data formats, connectors for every information system available on the market cannot be implemented in the first place. Instead the integration platform has to provide mechanisms that allow the cost efficient connection to a certain information system. This is where the strengths of the central component of the proposed software architecture come into play: an ESB supports multiple protocols (including a wide range of web services, REST, and other protocols), protocol conversion, data transformation, content-based routing, multiple connectivity options, and multiple standard business file formats for B2B communication [Fu09]. In addition the integration platform supports the propagation of business events to external subscribers via provided web services or via mail.

Information services: The integration platform is equipped with a relational database management system and thus provides reliable mechanisms to add, change, and delete information in a database and to query it for valuable information. Data on assets, smart object devices, and events are stored in the database. Since metadata on assets and smart object devices is very valuable this information is stored in the database too. For example in order to rate the quality of a sensor measurement in practice it is important to know some characteristics on the sensor that has measured a certain value. Thus for sensors the platform provides metadata like sensor type, measurement principle measured quantity, unit of measurement, measurement range, and the maximum sampling rate. Since assets strongly vary from application to application the attributes needed to describe these entities vary too. In the ideal case information on the asset can be retrieved from a connected enterprise information system. For situations where such information cannot be retrieved from an external source the platform provides a very flexible mechanism to define arbitrary assets: assets are stored according to the entity-attribute-value (EAV) representation that stems from medical informatics [Na99]. Additional information needed for asset matching (see asset management services) are retrieved from the ontology editor that is connected to the platform.

Furthermore the integration platform is equipped with a spatial database on top of the above mentioned relational database and a map server that can handle user-defined maps as well as user-defined areas and geo-fences within a map. Thus the platform provides geo-spatial queries like asking for additional information on the area that a certain asset is currently located and verifying whether a located asset has passed a virtual border (geo-fence) without permission. The platform implements a positioning engine that is responsible for calculating the location of smart object devices with appropriate location-sensing techniques [HB01]. Depending on the smart object technology a positioning engine can support different location models like presence-based locating, locating at room level, locating at choke points, locating by associating, and locating precisely [Ma09]. In fact the platform integrates a very simple positioning engine that does locating at choke points in order to compensate for smart object technologies like passive RFID that don't come with a positioning engine.

Finally the message store of the ESB is used in order to store logging data and to buffer information to be exchanged between different modules of the platform and between the platform and applications external to the platform. Thus the platform is able to bridge a temporary breakdown of a component.

Conversion services: Coordinates, units of measurement, and timestamps are stored and processed with unique data formats internally. Nevertheless the platform has to deal with several different formats for this kind of data. In practice different locating systems are deployed that use different coordinate systems. The same type of sensor can have different units of measurement (e.g. temperature can be measured in Celsius, Kelvin or Fahrenheit) and several different date and time notations are used around the world. Thus platform provides mechanisms that allow the conversion between these different data formats and the unique data formats used internally.

Security services: The platform supports security mechanisms like authentication and authorization for different users, groups of users, and roles of users. Again this is where the strengths of the central component of the proposed software architecture come into play: an ESB has integrated security features for authentication and authorization, supports access control lists (ACL), and allows services to be easily security enabled. In addition confidential data may be secured through the encryption and decryption module.

Maintenance services: Modules, interfaces between modules, and interfaces between modules and applications external to the platform are monitored by the platform. Corresponding logging data can be retrieved via the information services described earlier. Furthermore the platform provides mechanisms to configure the platform itself and connected smart objects devices. Maintaining a huge number of smart object devices can be a time-consuming task. For example in practice it might happen that firmware updates for all RFID interrogators of a certain category need to be deployed. Thus for smart object devices that can be updated remotely the platform supports remote firmware updates for a group of smart object devices that belong to the same category.

Decentralization services: Edge processing is a term that is often used in the context of RFID systems. The idea is to process and filter data as near as possible at the source of the data in order to achieve higher system scalability. In the field of WSN similar approaches are proposed that focus on decentralized in-network processing. For example since every wireless transmission consumes a significant amount of energy several approaches try to achieve higher sensor network lifetimes by reducing the amount of data to be transmitted. The idea is again to process and filter data as near as possible at the source of the data and to transmit aggregated data less often. From the authors point of view the general question what functionality should be centralized or decentralized and what technology should be used to shift functionality between the network and the backend is still not finally clarified yet. In the context of RFID mobile software agents might be a reasonable approach that enables the migration of functionality within the network. Intelligent RFID readers have a runtime environment like Java or .Net pre-installed and are thus well suited for modern software agent frameworks. Another interesting approach that should be considered is cloud computing [MG09].

Visualization services: The integration platform provides a web-based user interface that comprises a software dashboard, a display for digital maps, and an administration tool. A dashboard is useful for displaying the status of the components and interfaces of the platform as well as the status of defined key performance indicators (KPI) to be monitored by smart object technologies. Status information is displayed appropriately via reports, graphics, and gauges depending on the type of information. In addition there is a possibility to visualize tracking and tracing information of assets in a digital map by leveraging the geospatial information that can be retrieved from the information services. In practice it is necessary to display a ground plan of a certain building located at the enterprises premises and to model corresponding areas. For example different storage areas within a storage building might be displayed to a warehouseman. As stated in the description of the information services the platform supports user-defined areas and geofences. These can be easily defined by simply drawing polygons on a digital map. Finally a graphical administration tool is provided that allows the configuration of the platform, smart objects devices, assets, and of corresponding orders and tasks.

3.1 Relevant standards

Since the integration platform for diverse smart object technologies addresses existing enterprise infrastructures the platform needs to be standards compliant. Table 1 gives a brief overview on the platform modules and corresponding standards that were considered when implementing the needed functionality of a module. The list provided in Table 1 doesn't claim to be complete and reflects the authors' experiences. Furthermore details on mentioned standards are omitted since this would go beyond the scope of this paper. Nevertheless for additional information regarding RFID and sensor integration standards the interested reader may evaluate [Sá10].

Table 1: Overview on platform modules and corresponding standards

Module	Standards
Technology abstraction	WSN: IPv6, 6LoWPAN, OGC SWE; RFID: LLRP; RTLS: NMEA, GPX, KML
Filtering & enhancement	RFID: ALE, TDS, TDT
Event management	BRMS: SBVR, PRR, RuleML; PUBLISH-SUBSCRIBE: WS-Eventing, OGC SWE
Asset management	N/A
Visualization	JSF, Facelets, Ajax
Enterprise integration	CONNECTORS: JCA; B2B: EPCIS, EDI, ebXML, RosettaNet, HL7; MAIL: SMTP, IMAP, POP3; WEB SERVICES: SOAP over HTTP, WS-*, WSDL, XML, JBI
Information	DATA & METADATA: JPA, JDBC, SQL; GEOSPATIAL DATA: OGC WMS, OGC WFS, OGC WFS-T, OGC Simple Features; ONTOLOGY: OWL, RDF, SPARQL, SWRL
Conversion	DATE AND TIME NOTATION: UTC; COORDINATE SYSTEM: WGS84; UNITS OF MEASUREMENTS: SI
Security	JAAS, JCA/JCE, HTTPS, TLS
Maintenance	SNMP
Decentralization	SOFTWARE AGENTS: OSGi, FIPA; CLOUD COMPUTING: N/A

3.2 Functionalities required by different real world applications

In order to underpin that the functionalities proposed in section 3 stem from real applications of smart object technologies Table 2 compares different applications with these functionalities and lines out, which functionalities are required by each application. The applications of smart object technologies being compared are a WSN based system for asset management and transfusion safety in a clinical environment [Se09], a RTLS and WSN based system for material flow management in earthworks [Le10], and a RFID based system for the logistic traceability of turbine spare parts in the aviation industry [KL07].

Table 2: Functionalities required by different real world applications

Functionality	Clinical asset management and transfusion safety	Material flow management in earthworks	Traceability of turbine spare parts
Technology abstraction ...	●	●	●
for WSN	●	●	○
for RFID	○	○	●
for RTLS	(●)	●	(●)
Filtering & enhancement	●	●	●
Event management	●	●	●
Asset management	●	●	●
Visualization	○	●	●
Enterprise integration	●	(●)	●
Information	●	●	●
Conversion	●	●	(●)
Security	●	(●)	(●)
Maintenance	●	○	(●)
Decentralization	○	○	(●)

4 Related Work

The open source community provides a lot of solutions that were considered when implementing the proposed integration platform: Global Sensor Networks [AHS06] focuses on a sensor based abstraction of diverse TinyOS based sensor nodes, RFID readers, and cameras but doesn't consider RTLS or the integration of these sensors into existing infrastructures. Fosstrak [FRL07] aims at implementing the most important standards for RFID defined by EPCglobal, including LLRP, TDT, ALE, and EPCIS. WSN and RTLS are not addressed. ASPIRE [So09] focuses on RFID middleware and considers the EPCglobal standards ALE and EPCIS. The provision of an event management based on BRE and/or CEP seems to be considered as well as interfaces to ERP and WMS. With WSN another smart object technology is supported, but RTLS are not considered. Rifidi Edge [Sw09] is a RFID middleware that supports the EPCglobal standards LLRP and ALE as well as barcodes and arbitrary sensors, while RTLS are not addressed. Event management is implemented using CEP.

As can be seen from Table 1 in section 3.1 the OGC standard Sensor Web Enablement (SWE) might be an option that should be considered when addressing a technology abstraction as proposed in this paper. A detailed comparison of the EPCglobal Architecture Framework that comprises all standards of EPCglobal and SWE is given in [SH10]. An extension to the EPCglobal Architecture Framework named EPC Sensor Network that modifies ALE in order to support sensor values is presented in [SSK07].

When different RTLS should be supported over a common abstraction layer the dissertation of Jeffrey Hightower should be considered [Hi04]. This is also true for LocON [Co09]: Very interesting is the fact, that some of the quality indicators as proposed in this paper are considered since an indication of the expected quality of the location to high level applications is foreseen. Also information fusion for optimization of localization results seems to play a role.

The idea that commercial RFID middleware solutions should be further developed to support more than RFID is considered in the concepts Intelligent Network Sensor Infrastructure [SAW10] and Bloor Sensory Middleware architecture [Ho08]. Coming from RTLS in [Ma09] the key objective of RTLS middleware is defined as to make the applications independent of RTLS technology. Several technologies are mentioned that may be used to build a RTLS. This includes tailored RTLS solutions as well as passive and active RFID. Besides that the idea to integrate sensors in RTLS is mentioned only marginally.

5 Conclusion and future work

We showed that there is a need for an integration platform for diverse smart object technologies by motivating that the integration of diverse smart object technologies into existing enterprise systems is a complex, time-consuming, and expensive task. Furthermore we motivated that the consolidation of diverse smart object technologies demands new integration platforms and that using an integration platform for diverse smart object technologies is a prerequisite for achieving significant savings in integration costs. After answering what an integration platform for diverse smart object technologies is about we showed how such an integration platform should be designed and what functionalities are needed in order to be able to support heterogeneous applications of smart object technologies efficiently. We introduced an abstract software architecture for an integration platform for diverse smart object technologies and lined out, what functionalities such a platform should provide and what standards should be considered. Building on this we showed that the proposed functionalities conform to the requirements of different real world applications. Furthermore we compared our concept with related work and showed that to the best of our knowledge currently there is no solution available that meets all requirements that are proposed in this paper.

The modules proposed in the software architecture have been prototypically implemented in several different projects. At the moment the modules are tight together according to the proposed software architecture with an ESB. Since the platform is built on top of existing open source components it is very likely that we will release the platform under an appropriate open source license in the future. On the way to the first public version of the platform existing open source components and solutions will be further examined and analyzed. Finally we will evaluate our platform together with partners from industry and research.

Acknowledgements: This research is part of the projects Aletheia and Center for Intelligent Objects ZIO and was funded in part by the BMBF under grant number 01IA08001 and in part by the StMWIVT, respectively.

Bibliography

- [AHS06] Aberer, K.; Hauswirth, M.; Salehi, A.: Global Sensor Networks. Technical Report LSIR-REPORT-2006-001, Lausanne, Switzerland, 2006.
- [AP03] Alvarez, C. E.; Pardue, H.: A Reference Architecture based on Web Components for Ubiquitous Information Systems: AMCIS 2003, Tampa, Florida, USA, 4-5 August 2003, pp. 1901–1910.
- [Br06] Brodie, M. L.: Integration in A Service-Oriented World. The Big Picture., I-ESA 2006, Bordeaux, France, 2006.
- [Co09] Couronné, S. et al.: LocON - a Platform for an Inter-Working of Embedded Localisation and Communication Systems. Poster Abstract. In (IEEE Computer Society Ed.): IEEE Secon 2009.
- [De10] Defining Technology Inc.: Middleware Resource Center. What is Middleware? <http://middleware.org/whatis.html>, accessed 4 Oct 2010.
- [De82] DeMarco, T.: Controlling software projects. Management, measurement and estimation. Yourdon Press, NY, NY, USA, 1982.
- [Di09] Dienelt, S.: EPC/Rfid und Sensorik. Grundlageninformationen, 2009.
- [Fi08] Finkenzeller, K.: RFID-Handbuch. Hanser, Munich, Germany, 2008.
- [FRL07] Floerkemeier, C.; Roduner, C.; Lampe, M.: RFID Application Development With the Accada Middleware Platform. In IEEE Systems Journal, 2007, 1; pp. 82–94.
- [Fu09] Fulton, L. et al.: The Forrester Wave: Enterprise Service Buses, Q1 2009. Technical Report, Cambridge, MA, USA, 2009.
- [Ga93] Garvin, D. A.: Building a Learning Organization. In Harvard Business Review, 1993, 71; pp. 78–91.
- [Gu09] Gualtieri, M. et al.: The Forrester Wave: Complex Event Processing (CEP) Platforms, Q3 2009. Technical Report, Cambridge, MA, USA, 2009.

- [HB01] Hightower, J.; Borriello, G.: Location Systems for Ubiquitous Computing. In IEEE Computer, 2001, 34; pp. 57–66.
- [Hi04] Hightower, J.: The Location Stack. Dissertation, Seattle, Washington, USA, 2004.
- [Ho08] Holloway, S.: RFID Middleware. From RFID to Sensory Network middleware f. Technical Report, Towcester, UK, 2008.
- [Ho09] Holloway, S.: Business Rules Management. Managing business rules of an organisation. Technical Report, London, UK, 2009.
- [KL07] Krupp, M.; Lempert, S.: Prozesskompetenz als Basis für erfolgreiches Event Management, SCEM Forum 2007, Wiesbaden, Germany, 2007.
- [KW07] Karl, H.; Willig, A.: Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons, Chichester, West Sussex, UK, 2007.
- [Le04] Leaver, S. et al.: Evaluating RFID Middleware. Technical Report, Cambridge, MA, USA, 2004.
- [Le10] Lempert, S. et al.: Transparente und effiziente Prozesse im Erdbau durch ereignisgesteuertes Stoffstrommanagement auf Basis von Smart Objects und Business Rule Management. In (Fährnich, K.-P.; Franczyk, B. Eds.): INFORMATIK 2010, Leipzig, Germany, 27. September - 1. October 2010. GI, Bonn, 2010; pp. 207–212.
- [Li99] Linthicum, D. S.: Enterprise Application Integration. Addison-Wesley, Boston, MA, USA, 1999.
- [LLS06] Laudon, K. C.; Laudon, J. P.; Schoder, D.: Wirtschaftsinformatik. Eine Einführung. Pearson Studium, Munich, Germany, 2006.
- [Ma09] Malik, A.: RTLS for Dummies. Wiley, Indianapolis, IN, USA, 2009.
- [Ma69] Mackenzie, R. A.: The Management Process in 3-D. In Harvard Business Review, 1969, 47; pp. 80–82.
- [MG09] Mell, P.; Grance, T.: The NIST Definition of Cloud Computing. Version 15, 2009.
- [Mo09] Mojix Inc.: Mojix EPC Compliant Real-Time Location System. www.mojix.com/products/documents/RTLS_brochure.pdf, accessed 4 Oct 2010.
- [Na99] Nadkarni, P. M. et al.: Organization of Heterogeneous Scientific Data Using the EAV/CR Representation. In JAMIA, 1999, 6; pp. 478–493.
- [NLF07] Nakamura, E. F.; Loureiro, A. A. F.; Frery, A. C.: Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications. In ACM Computing Surveys, 2007, 39; Article No. 9.
- [NR68] Naur, P.; Randell, B. Eds.: NATO Software Engineering Conference 1968. Garmisch, Germany, 7 - 11 October 1968, 1968.

- [OA06] OASIS: Reference Model for Service Oriented Architecture 1.0, 2006.
- [OA09] OASIS: Reference Architecture Foundation for Service Oriented Architecture Version 1.0, 2009.
- [Ra08] Rainer, F. et al.: From Academia to the Field. Wireless Sensor Networks for Industrial Use. In (Ritter, H. et al. Eds.): FGSN 2008. Berlin, Germany, 25 - 26 September 2008; pp. 93–95.
- [Ri10] Ritchie, P.: DASH7 Alliance Announces Updated Standard For Wireless Sensor Networks. San Ramon, CA, USA, 2010.
- [RP07] Ruser, H.; Puente León, F.: Informationsfusion - Eine Übersicht. In Technisches Messen, 2007, 74; pp. 93–102.
- [Sá09] Sánchez López, T. et al.: Taxonomy, technology and applications of smart objects. In Information Systems Frontiers, 2009.
- [Sá10] Sánchez López, T.: RFID and sensor integration standards. In Computer Standards & Interfaces, 2011, 33; pp. 207-213.
- [SAW10] Sirico, L.; Arteaga, C.; Woods, T.: RFID Middleware is Extinct: The Intelligent Sensor Network is Born. <http://rfid.net/basics/middleware/143-rfid-middleware-is-extinct-the-intelligent-sensor-network-is-born>, accessed 15 Oct 2010.
- [Se09] Sedlmayr, M. et al.: Towards a Smart Object Network for Clinical Services: AMIA 2009. San Francisco, CA, USA, 14 - 18 November 2009, 2009; pp. 578–582.
- [SH10] Sánchez López, T.; Harrison, M.: Supply Chain sensor support by integrating the OGC Sensor Web Enablement and the EPC Network architectures. White Paper SWSNET-028, Cambridge, UK, 2010.
- [So09] Soldatos, J.: AspireRFID Can Lower Deployment Costs. <http://www.rfidjournal.com/article/view/4661>, accessed 4 Oct 2010.
- [SSK07] Sung, J.; Sánchez López, T.; Kim, D.: The EPC Sensor Network for RFID and WSN Integration Infrastructure. In (Hurson, A.; Pingali, G. Eds.): PerCom workshops 2007. White Plains, NY, USA, 19 - 23 March 2007. IEEE Computer Society, Los Alamitos, CA, USA, 2007; pp. 618–621.
- [Sw09] Swedberg, C.: Pramari Launches Free Open-Source RFID Middleware. <http://www.rfidjournal.com/article/view/5328>, accessed 4 Oct 2010.
- [Th06] Thomas, O.: Understanding the Term Reference Model in Information Systems Research. In (Bussler, C.; Haller, A. Eds.): BPM 2005. Nancy, France, 5 September 2005. Springer, Berlin, 2006; pp. 484–496.
- [Th09] The Open Group: SOA Reference Architecture, 2009.
- [VD10] Vasseur, J.-P.; Dunkels, A.: Interconnecting smart objects with IP. The next Internet. Morgan Kaufmann/Elsevier, Burlington, MA, USA, 2010.