Scalable Computing: Practice and Experience
Volume 12, Number 3, pp. 307–315. http://www.scpe.org
ISSN 1895-1767

THE SEMANTIC MIDDLEWARE FOR NETWORKED EMBEDDED SYSTEMS APPLIED IN THE INTERNET OF THINGS AND SERVICES DOMAIN

PETER KOSTELNÍK[†], MARTIN SARNOVSKÝ[†], AND KAROL FURDÍK[†‡]

Abstract. The paper presents the LinkSmart middleware platform that addresses the Internet of Things and Services approach. The platform was designed to support the interoperability and seamless integration of various external devices, sensors, and services into the mainstream enterprise systems. The design and development of LinkSmart goes across two integrated European research projects, namely the FP6 IST project Hydra and the FP7 ICT project EBBITS. Modular architecture and functionality of LinkSmart prototype, developed by combining the service-oriented architecture, peer-to-peer networking, and semantic web services technologies, is described with focus on semantic binding of networked devices by means of underlying ontologies and knowledge-based inference mechanisms. Extensions of the solution towards the service orchestration, complex event handling, business process modelling and workflow processing are discussed and described on a mechanism of context-aware processing of sensor data.

Key words: semantic web services, Internet of Things and Services, ontology for services, devices and events, networked embedded systems

AMS subject classifications. 94-04, 94B99

1. Introduction. The innovative and rapidly evolving research area of *Internet of Things and Services* (IoTS) [16], [20] addresses an investigation of ways and means for seamless functional interconnection and effective, so-called intelligent, communication of various devices, services, information systems and resources towards operational scenarios. The aim of efforts in this area is focused on a development of platforms and solutions providing pervasive computing environment for networked embedded systems [6], which may be employed in real-world applications in industrial domains such as manufacturing, e-Business, e-Health, etc. In the European context, the IoTS research is supported, for example, by the CERP-IoT cluster [20], which helps to co-ordinate the research efforts in tens of involved FP6/FP7 projects. The projects such as ASPIRE, BRIDGE, CoBIs, CuteLoop, Hydra, and EBBITS can be explicitly mentioned as mostly related to the topic of this paper, i.e. the IoTS-enabling middleware. A brief survey of research activities, approaches, and technology solutions in this field will be presented in Section 1.1. The approach towards the IoTS middleware, which will be specifically discussed and presented later in this paper, was proposed and elaborated within the Hydra and EBBITS projects, where the authors of the paper were involved as development partners.

A wide scale of technologies is employed in IoTS frameworks or applications, ranking from Radio Frequency Identification (RFID) and sensor signals processing to computer network technologies, web services, or Service Oriented Architectures (SOA) of information systems. However, here we will focus on the middleware layer that enables, by means of semantic web services and related knowledge representation structures, a networking of physical devices, sensors, or components in order to provide higher value-added (i.e., more advanced, sophisticated, or intelligent) solutions to the end users. This was the objective of the FP6 integrated project Hydra (*Networked Embedded System Middleware for Heterogeneous Physical Devices in a Distributed Architecture*, http://www.hydramiddleware.eu), which started in July 2007 and finished in December 2010. Hydra aimed at the development of a middleware for intelligent networked embedded system, which is based on service-oriented architecture and is deployable on both new and existing networks of distributed wireless and wired devices [3]. The resulting system, which was named as the *LinkSmart* middleware, is described in Section 2.

Furthermore, the development continued by adapting the middleware for a broader exploitation. Directions of extensions and enhancements were identified namely in the underlying semantic structures, where several significant improvements were proposed - for example, a new ontological model of generated events, more advanced reasoning, inclusion of more types of devices, etc. To test these extensions, two application cases were specified in the areas of automotive manufacturing and food traceability. A new FP7 integrated project EBBITS (*Enabling Business-Based Internet of Things and Services*, http://www.ebbits-project.eu) [21] was launched in September 2010, where the major part of the Hydra consortium decided to participate. EBBITS is coordinated by the Fraunhofer Institute FIT (Institute for Applied Information Technology, http://www.fit.fraunhofer.de), which formerly coordinated the Hydra project as well. EBBITS consortium includes two industrial partners that

[†]Technical University of Košice, Faculty of Electrical Engineering and Informatics, Letná 9, 042 00 Košice, Slovakia. (Peter.Kostelnik@tuke.sk, Martin.Sarnovsky@tuke.sk, Karol.Furdik@tuke.sk).
[‡]InterSoft, a.s., Floriánska 19, 040 01 Košice, Slovakia. (Karol.Furdik@intersoft.sk).

are responsible for setting up and running pilot applications in automotive manufacturing and food production domains. Furthermore, seven research institutes and universities are the project partners that provide research, system design, development, and implementation work. Home institutions of authors of the paper, i.e. Technical University of Košice and InterSoft, a.s., are mostly involved in technology-related investigations of IoTS, design of the EBBITS system architecture, as well as in the design and implementation of semantic structures and supportive software components required for extending the LinkSmart middleware towards the advanced service/sensor data fusion and integration of these low-level data into business process workflow sequences.

The rest of the paper is organised as follows. The next subsection provides an overview of related technologies and research projects, which were investigated as background knowledge resources for both Hydra and EBBITS projects. Similarities, distinctions, and advancements of the Hydra / EBBITS approach in a comparison to the listed projects are briefly discussed. Section 2 presents the functionality and architecture of the LinkSmart middleware. General description of the system [3, 17] is accompanied by a description of using semantic infrastructure of LinkSmart for binding physical devices into a broader IoT network, which is demonstrated on a simple example. The semantic model of services, which can be considered as one of key conceptual structures in the LinkSmart infrastructure, is described in Subsection 2.2. The OWL ontology of services is presented together with the updates that we were already designed for applying the LinkSmart middleware in the IoTS platform of EBBITS. An overview of LinkSmart enhancements that will be accomplished in EBBITS is included in Section 3. Designed system architecture is presented together with the ontology of events, specifically developed for the purposes of complex event processing and inclusion of semantically enriched sensor data to the respective business processes. Section 4 summarises the adopted approach of IoTS, outlines planned pilot applications of EBBITS, and presents planned steps of further research.

1.1. Related Technologies and Research Streams. The area of IoTS is related to several research and application domains, which include semantic software architectures and ontologies, semantic web services, knowledge systems, middleware platforms, smart sensor networks, as well as a variety of systems providing features of distributed intelligence and/or content awareness for devices. A survey of enabling technologies, SOA-based middleware solutions, and IoTS applications can be found, for example, in [1]. Authors of this article emphasise the interoperability of interconnected devices as a central issue of IoTS, which includes "*an always higher degree of device smartness by enabling their adaptation and autonomous behaviour, while guaranteeing trust, privacy, and security*". The interoperability in its technical, semantic, and organisational aspects can be achieved by proper communication and networking infrastructure, adoption of semantic technologies, and integration of semantically annotated device events to complex workflow structures of high-level business processes, respectively.

In the IoT/IoTS middleware systems, the communication infrastructure is typically (but not exclusively [5]) based on SOA principles and peer-to-peer (P2P) networking [13], which is enabled by technologies such as JXTA (http://jxta.kenai.com), Java Message Service, event processing network [4], etc. To integrate the wireless sensor networks in the P2P platform by means of device proxies, the Contiki platform (http://www.contiki-os.org) can be employed to run the messages in 6LoWPAN standard (http://datatracker.ietf.org/wg/6lowpan/), which is based on IEEE 802.15.4-2003 standard. These technologies were proposed for EBBITS to enable the opportunistic sensor and device networking, which can be considered as a step towards the technical interoperability.

The semantic interoperability is achieved by employing technologies such as ontologies and knowledge management systems that facilitate logical reasoning, clustering of sensor data to more complex events, mediation of semantically heterogeneous data or interfaces, decision making and context awareness for networked sensors, devices, and services. Context-aware middleware characteristics and application types were analysed in [9], together with a benchmark of nine middleware systems that were produced mostly as outcomes of various research projects such as Aura, CARISMA, CORTEX, SOCAM, etc. [18]. This analysis served as a starting point for Hydra, which was focused on the development of SOA-based middleware providing a transparent communication layer for embedded devices. Special emphasis was given on features such as the interoperable access to data, information and knowledge across heterogeneous platforms, including web services, and support true ambient intelligence for ubiquitous networked devices [11, 3].

In the area of semantic web and knowledge systems, EBBITS will use knowledge engineering methods and tools to extend the Hydra ontologies and to develop advanced reusable domain models in the scope of pilot enterprise applications. From a scale of relevant approaches, we can mention two FP6 integrated projects:

- *SEKT* (http://www.sekt-project.com), which developed and exploited semantic knowledge technologies of ontologies, metadata, and knowledge discovery. In EBBITS, we plan to re-use and extend these technologies towards the detection of useful complex features that can help in decision making process.

- *DIP* (http://dip.semanticweb.org), providing an infrastructure for semantic web services to e-Work and e-Commerce, is relevant to the further development of specific middleware services in EBBITS.

Since the EBBITS platform focuses on an effective applicability in a real-word industrial environment, the criteria such as robustness, usability, and scalability are of high importance. Based on a selection of the well-proven RDF/OWL knowledge representation formalisms and related reasoning mechanisms [7], the BigOWLIM[1] and AllegroGraph[2] storage platforms for RDF triples were identified as the best candidates to provide effective storage and access facilities for the middleware semantic data [2, 10].

Real word enterprise applications of IoTS middleware require a strong support of process workflows and related semantic web services, which addresses the organisational aspect of interoperability. On the side of services, it requires capabilities for discovery, composition (i.e. orchestration or choreography), deployment, and execution of services in a pre-defined or ad-hoc created workflow [22]. The workflow sequences can be modelled as abstract business processes, represented by the BPMN 2.0 notation (http://www.bpmn.org), which can be transformed to the respective executable form. From several available solutions we have selected the Drools platform (http://www.jboss.org/drools), which employs jBPM 5 toolkit for maintenance of BPMN process models combined with business rules (cf. Section 3).

Projects and approaches in the area of IoTS middleware, which were identified as the most relevant for the design and development of the EBBITS platform, are as follows:

- *ASPIRE* (http://www.fp7-aspire.eu), an FP7 project that has designed and developed a lightweight, royalty-free, and integrated middleware platform that could be used to implement the RFID identification part of the EBBITS system [8]. In an opposite way, EBBITS solution on distributed intelligence and semantic knowledge infrastructure could help ASPIRE to extend the architecture.
- *BRIDGE* (http://www.bridge-project.eu), an FP6 project that provides a suite of RFID tools and business cases that could be employed namely in the food traceability scenario of EBBITS pilot application. In addition, BRIDGE tools enable handling of the Electronic Product Code standard (EPC, http://www.epcglobalinc.org), that will be taken as one of resources to extend the Hydra device ontologies towards the EBBITS pilots.
- *SENSEI* (http://www.sensei-project.eu), an FP7 project that was aiming to create a business driven, scalable, pluggable and open framework for heterogeneous wireless sensor and actuator networks [19]. However, the project mainly addresses the scalability issue and the definition of services interfaces, which need to be extended in EBBITS by semantic-oriented integration capabilities and distributed intelligence features.

Other related approaches and solutions, as well as visions and challenges for the IoTS domain, are investigated, developed and provided as outcomes of projects co-operating within the CERP-IoT cluster [20], where both Hydra and EBBITS projects are included.

2. Architecture and Functionality of the LinkSmart Middleware. The Hydra project, briefly introduced in Section 1, was aimed at research, development, and validation of a middleware for networked embedded systems that would allow a development of cost-effective, high-performance ambient intelligence applications for heterogeneous physical devices [3]. To test the solution in a real-world environment, three pilot applications were prepared and accomplished in domains of facility management (smart homes), healthcare, and agriculture.

2.1. Semantic Binding of Devices in LinkSmart. The LinkSmart middleware, produced as the main outcome of the Hydra project, combines the semantic web services technology with SOA-based principles applied on the solution. The SOA and its related standards provide interoperability at a syntactic level. However, Hydra also aims to provide interoperability at the semantic level. One of the objectives is to extend the syntactic interoperability to the application level in terms of semantic interoperability. This was accomplished by combining the use of ontologies with semantic web services. In this context, Hydra introduces the *Semantic Model Driven Architecture* (SeMDA) [15], which was designed to facilitate an application development and to promote semantic interoperability for on-line services and devices of wireless or wired type [3]. The SeMDA of Hydra includes a set of ontologies, and provides the set of tools, which can be used both in application design

[1]http://www.ontotext.com/owlim/
[2]http://www.franz.com/agraph/allegrograph/

time and runtime [12]. The SeMDA concept, implemented in the LinkSmart middleware, makes all devices in a LinkSmart-based IoTS application accessible in an uniform way - as the semantic web services.

Basically, SeMDA in LinkSmart provides a mechanism for wrapping standard API interfaces of services, sensors, and various physical devices with a defined web service extension, which is enhanced by a semantic description of provided or generated WSDL files [6]. This process is called the Hydra-enabling of the device. Developer can Hydra-enable a new device using so-called Device Development Kit (DDK), included into the LinkSmart infrastructure [17]. The new device is annotated to the suitable class in the device taxonomy (e.g. mobile device) and the basic description, such as device model name and number, manufacturer information, energy consumption profile or device discovery information, is added. Since particular devices may have different connection and communication capabilities, the service calls have to be transformed into web service calls. For each service, the developer has to add a custom implementation, which includes common services as *StartWS*, *StopWS* or *GetWSEndpoint*, and services related to the energy consumption such as *CurrentPowerConsumption* or *RemainingBattery*. Each service is also annotated to the suitable service taxonomy class of LinkSmart Device ontology. This way, the devices and their local networks are both accessible by LinkSmart and connected to the outside world through broadband and/or wireless networks. For example, the binding of a thermometer device to the respective semantic description in Device.owl ontology of LinkSmart is as follows:

```
<linksmart:binding device="http://linksmart.eu.com/
  ontology/Device.owl#thermometer"/>
```

The LinkSmart can generate a stub of the related client and/or server code, which can be based, for example, on an available ontology instance that semantically describes the states of the device or sensor. The proxy stub is created according to the devices capabilities as either directly embedded on the device or using the *OSGi framework*[3]. The device can then be accessed and controlled in the application code of a networked embedded system using the following Java statements:

```
AppDeviceManager myMgr = new AppDeviceManager();
ThermoMeter.LSDeviceWS myThermometer = new ThermoMeter.LSDeviceWS();
myThermoMeter.SetLSID(myMgr.GetLSID("Off1Thermometer"));
Light.LSDeviceWS myLight = new Light.LSDeviceWS();
myLight.SetLSID(myMgr.GetHID("Off1Light"));
...
if (myThermometer.GetTemperatureC() > 25)
{
  myLight.Flash(2);
  myPhone.SendSMS("Too hot in the office 1,
    temperature:" + myThermometer.GetTemperatureC()+ "+421329264552");
}
```

The meaning of the presented code, which is rather simplified for demonstration purposes, is as follows. After initiating the Application Device Manager object (which is included in the Service Layer of Application Elements, as it is presented in Figure 2.1), a web service client is created for the thermometer device. It is required that the device is Hydra-enabled, so that it was properly wrapped by obligatory web service interfaces and annotated by LinkSmart device ontology. Then the LinkSmart identifier, abbreviated as LSID, is retrieved from a concrete physical device in our case, from the thermometer located in an office 1 (i.e. from Off1Thermometer). The identifier is used to create an endpoint URL for the device, forwarded as input parameter to the web service client that corresponds to the device. The same way, another web service client is created and initialised for the light device. Once the web service client was initialised and the URL for a device was properly established, the customised LinkSmart services of the device can be invoked and consumed. The rest of the code is obvious if the temperature in office 1, measured continuously by the thermometer, exceeds 25 degrees, then the light will start flashing and the cell phone will send a message.

The creation of a new Hydra-enabled device in design time introduces only basic device semantic representation, which can be later further extended. Each device ontology instance represents the specific device model and serves as the static information template [11]. In runtime, when new device enters in the LinkSmart

[3] http://www.osgi.org

application network, the best matching template is identified by the semantic discovery process, cloned and tied to the physical device using the LSID persistent identifier. The property values of the runtime instance can change as the device changes its state variables (e.g. measured values of thermometer or sensor). When physical device leaves the LinkSmart network, assigned device runtime instance is removed from the ontology.

2.2. LinkSmart Architecture and Components. LinkSmart middleware is typically installed as a node in the peer-to-peer network, which encapsulates interfaces of internally referenced devices and provides them as semantic web services to other network nodes - LinkSmart instances. The architecture of the main functional modules of LinkSmart is depicted in Figure 2.1.

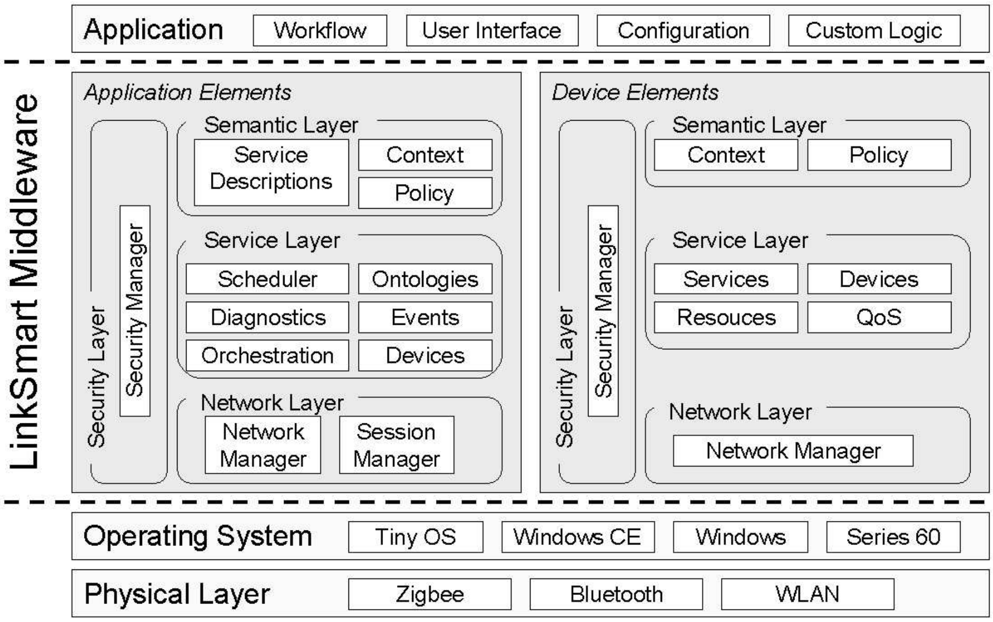

Fig. 2.1: Structural overview of the LinkSmart middleware layers

The inner middleware elements are enclosed by the physical, operating system and the application layers shown at the bottom and at the top of the diagram, respectively. The physical layer provides several network connections like ZigBee, Bluetooth or WLAN. The operating system layer enables managing the physical layer objects and provides methods for accessing the resources of network connections. The operating system layer provides means to access and manage the physical layer objects. The application layer contains customisable user applications that may include modules for workflow management, user interface, custom logic and configuration details. These three layers are not a part of the LinkSmart middleware.

The middleware itself is divided into the *Application Elements* and *Device Elements* parts, representing the close (i.e. running in a performance-wise mode, e.g. on the same machine as the resources used) and distant (i.e. remote, with a slow access or performance) components, respectively. Each of the parts consists of the network, service, semantic, and security layers, which contain the LinkSmart business logic, i.e. the functions for context sensing, service requests handling, network management and synchronisation of peer nodes, access control, etc.

The middleware functionality is supported by a structure of OWL ontologies, which provide a semantic basis for particular business logic elements. For example, the ontology structure of the LinkSmart service model is presented in Figure 2.2. Services, which are tied to devices, are described by the respective capabilities, input and output parameters such as name, data type, and unit. Similar ontologies were produced for modelling devices, network connections, and security issues [12].

3. EBBITS Extensions towards the IoTS Domain. The EBBITS project is aiming to shift the IoTS paradigm of Hydra more towards the services that are orchestrated in complex workflow sequences, i.e. in business processes that correspond to the real-world scenarios in industry or other application domain [21].

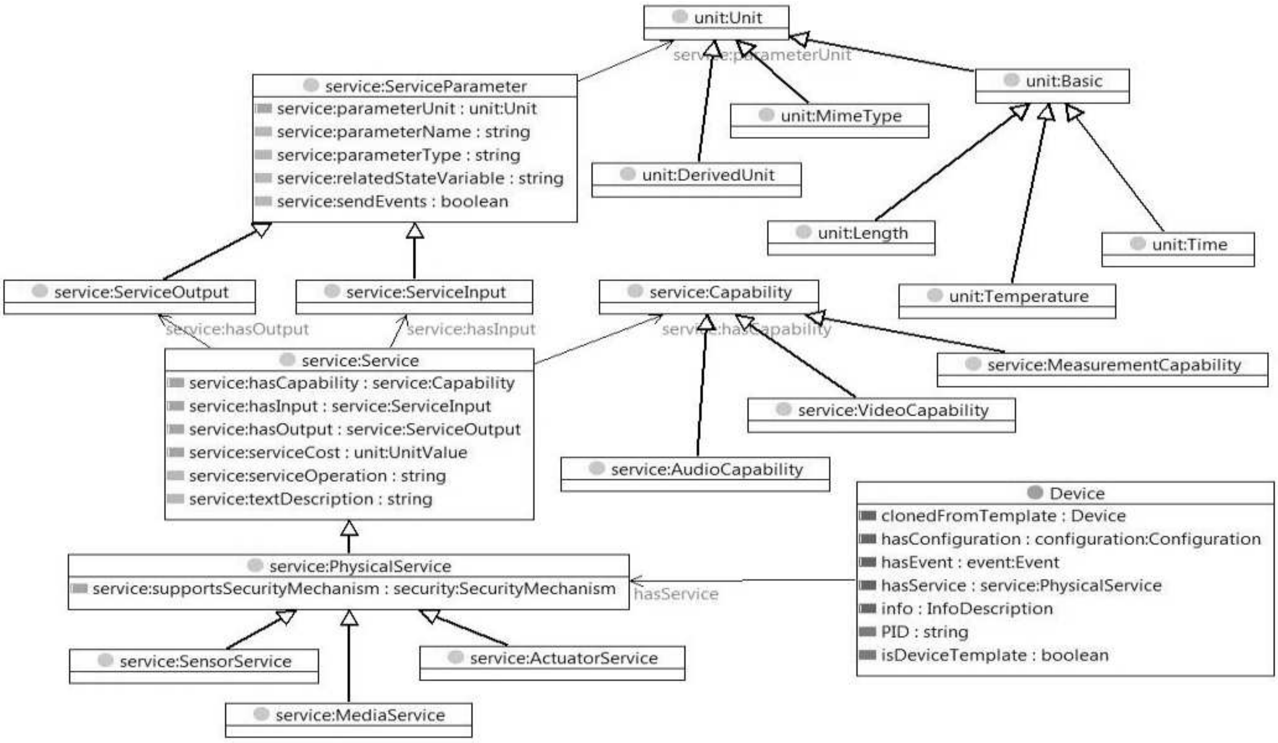

Fig. 2.2: Conceptual model of services in the LinkSmart ontology

This way, the EBBITS platform should provide a bridge between enterprise and public information systems, as well as between human users and things in the physical world.

The EBBITS platform is targeting to support interoperable business applications with context-aware processing of data separated in time and space, information and real-world events (addressing tags, sensors and actuators as services), people and workflow chains (operator and maintenance crews). Optimisation of service compositions will be supported by means of high level business rules that can be driven, for example, by energy or cost performance criteria. The key requirement for the business rules execution is that the EBBITS platform needs to be able to recognise and respond to physical world events. The information acquired from events, which may come from physical world and are generated by various devices, create the basis for decision making at the several levels of the EBBITS architecture (cf. Figure 3.1), including data fusion, situation patterns recognition, complex event processing, analysis of historical acquired data, etc. All these requirements need to work with a large amount of information related to the devices generating events or providing services for further processing by event/service orchestration, decision making, or business rules. In some cases it must be possible to use this information to analyse the historical data generated by particular events. All parts of decision making process will be supported by enriched semantic model that will enable a flexible knowledge representation of all included events, roles, services and processes.

Obviously, EBBITS builds on the outcomes of the Hydra project. The LinkSmart middleware system is taken as the implementation basis, which will be extended on the functions and capabilities of semantic business process modelling, workflow management, service choreography and orchestration, event handling and processing of complex events generated by devices. The development will also address the service interoperability issues and various enhancements that can be required on the security and networking maintenance.

A high-level architecture of the main functional modules proposed for EBBITS is presented in Figure 3.1. Physical level of devices, sensors, external services or applications is constituted on the same principles as in LinkSmart solution. It means that the devices are included into the LinkSmart application network by the semantic binding mechanism and supported web service interfaces. Devices may generate events, which are collected on the *Physical World Adaptation Layer*. After a normalisation and resolving of initial semantic

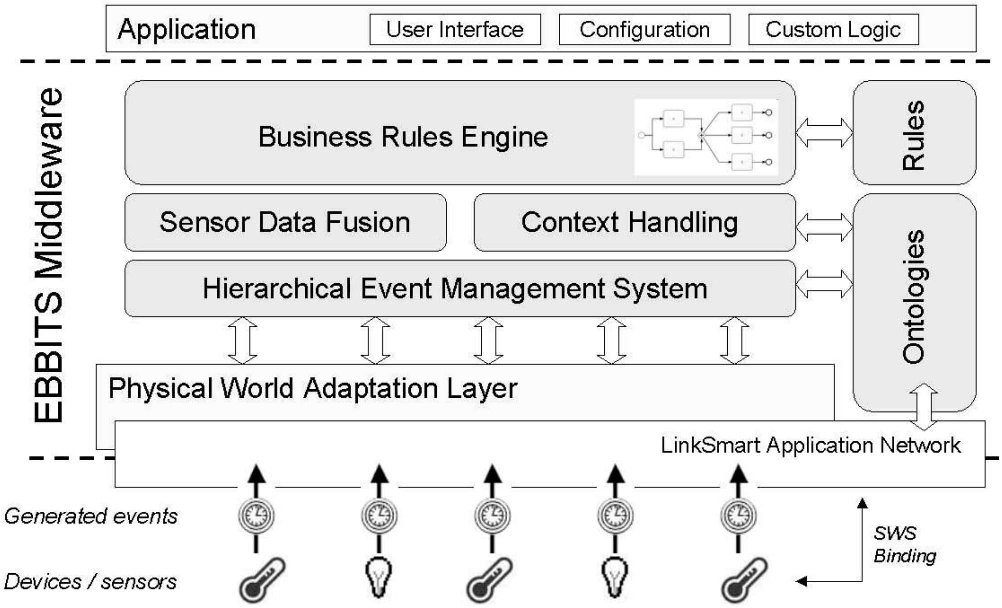

Fig. 3.1: Architecture of functional modules for the EBBITS platform

characteristics (using, for example, classification, pattern recognition, or clustering methods), the events are propagated to the *Hierarchical Event Management System* (HEMS). The *Sensor Data Fusion* module, invoked by HEMS, will combine low-level sensor events into more complex information structures. Semantic search, mediation, and reasoning mechanisms, supported by applied RDF triple store framework and related reasoning engines [10], will be employed for sensor data merging and context handling. Consequently, the data fusion should provide information chunks that are suitable for further processing by workflow elements and business rules of IF-THEN format, provided by the jBPM5 toolkit included into the Drools platform (see Section 1.1).

Communication and data/information transformation is expected in both directions. If the data flow goes from events to business rules, then the execution of a pre-defined business process is driven, or at least influenced, by generated events. In case of opposite data flow, a network of devices can be controlled and driven by business rules specified on the upper level of EBBITS middleware.

As one of initial steps towards the extensions required on LinkSmart system for providing the functionality proposed for EBBITS, it was necessary to update the semantic model for events, which has originally been developed in LinkSmart on an insufficient level of details. We have designed the events ontology that should cover all the information related to the hierarchical event handling and sensor data fusion in EBBITS, as it is depicted in Figure 3.2. The ontology was created with respect to the *SSN ontology* [14]. It includes a generic Event concept, models of event results and event stimulus, as well as the connection to the service that triggers the event. These features should support the data fusion by, for example, merging of events generated by devices of the same type, location, etc. The core taxonomy of events, required for hierarchical event management, consists of two main sub-classes, which are distinguished according to the event stimulus type, i.e., triggered by a real-world situation or continually generated in some frequency.

Further EBBITS enhancements of LinkSmart will include, among others, a semantic model for service composition, which will cover the service execution preconditions and post-conditions, the models for orchestration of services into processes or grounding the services to a concrete implementation. The design of these extensions will be most likely driven by OWL-S, WSMO, or similar semantic service ontologies.

4. Conclusions. To summarise, the Hydra and EBBITS projects present the concept of Semantic Devices [11]. The motivation behind the concept is the fact that the services offered by physical devices are generally designed independently of the particular applications in which the devices might be used. A semantic device, on the other hand, represents what a particular application would like to have. The basic idea of this approach is to hide all the underlying complexity of the mapping to, discovery of, and access to physical devices. The

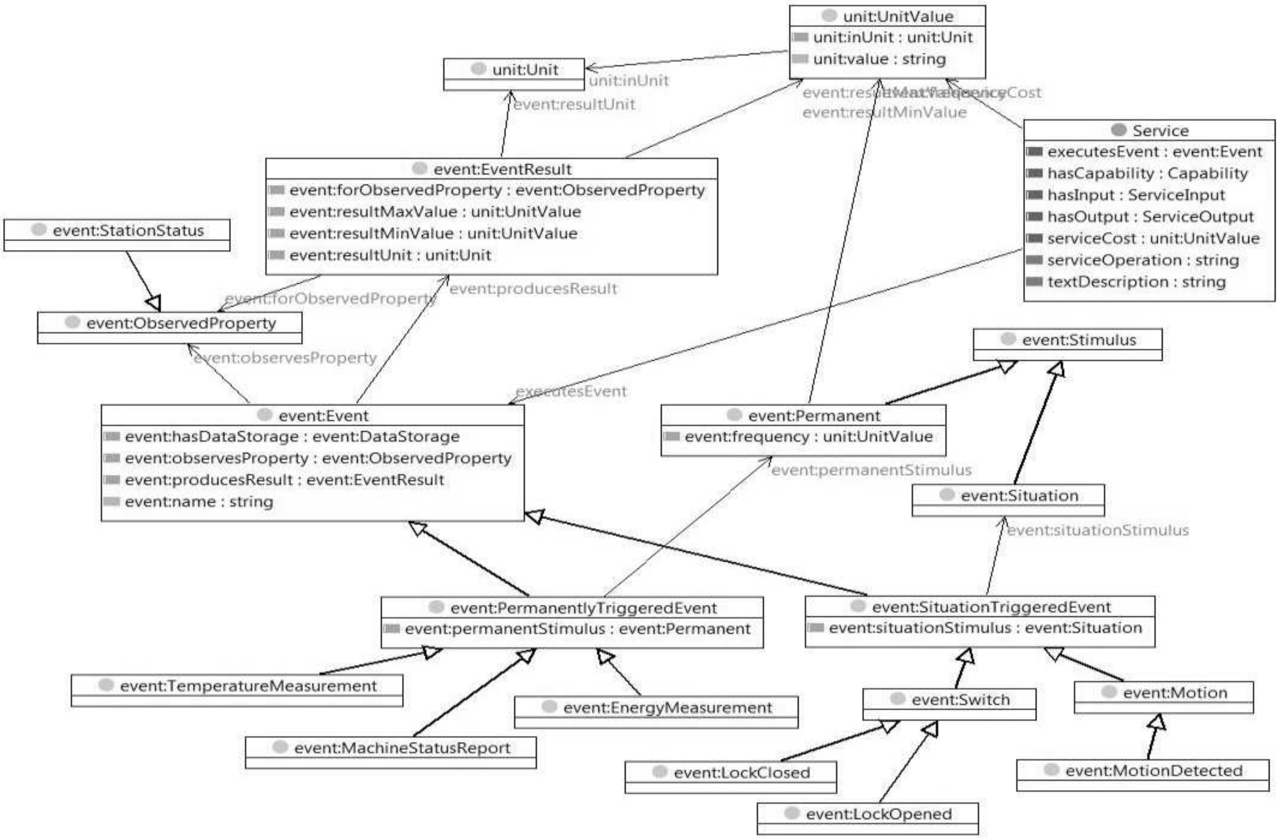

Fig. 3.2: Conceptual model of events proposed for the EBBITS platform

programmer just uses it as a normal object in his application focusing on solving the applications problems rather than the intrinsic of the physical devices.

The foundations of this concept were investigated in the Hydra project and resulted in the LinkSmart middleware system, currently available under the LGPL 3.0 open source licence. The EBBITS project builds on the Hydra outcomes and tries to extend the LinkSmart software on a fully featured support of business processes, event handling, and workflow processing. This approach will be tested in two pilot applications, namely:

- *Automotive manufacturing*, where enhanced LinkSmart middleware should help to manage the production optimisation with special focus on reducing energy consumption;
- *Food producing industry*, supporting the service and resource traceability through the whole process cycle, from the production to consumption stage.

Currently, in July 2011, the design and analysis work is completed and the specification of LinkSmart system updates is ongoing. Technologies and related approaches were identified for components and semantic structures that will be redesigned or newly developed within EBBITS. Specification of envisioned updates is available and the implementation is ongoing. The first prototype of the enhanced LinkSmart system should be available in autumn 2011.

Acknowledgments. The project Hydra was co-founded by the European Commission within the 6th Framework Programme, contract No. IST-2005-034891. The presented work is partially funded by the European research project EBBITS (7th Framework Programme, contract No. ICT-2009-5-257852), and by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic within the 1/0042/10 Project "Methods for identification, annotation, search, access and composition of services using semantic metadata in support of selected process types". We like to thank all partners in the EBBITS project for their support.

REFERENCES

[1] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A survey, in Computer Networks, The International Journal of Computer and Telecommunications Networking, Volume 54 Issue 15, October, 2010, pp. 2787–2805.

[2] C. Bizer, A. Schultz, *Berlin SPARQL Benchmark Results*, Freie Universität Berlin, 2009, [on line], [cit. August 3, 2011], available at http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/results/index.html.

[3] M. Eisenhauer, P. Rosengren, P. Antolin, *HYDRA: A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems*, in The Internet of Things, Springer, New York, 2010, pp. 367–373.

[4] O. Etzion, P. Niblett, *Event Processing in Action*, Manning Publications Co., Greenwich, CT, USA, 2010.

[5] C. Floerkemeier, C. Roduner, M. Lampe, *RFID application development with the Accada middleware platform*, in IEEE System Journal 1 (2), 2007, pp. 82-94.

[6] K. M. Hansen, W. Zhang, G. Soares, *, G.: Ontology-Enabled Generation of Embedded Web Services*, in Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering, Redwood City, San Francisco Bay, USA, 2008, pp. 345–350.

[7] J. Hreno, P. Nutakki, M. Knechtel, K. Furdik, T. Sabol, *Deliverable 4.2: Knowledge representation formalism analysis*, Project Deliverable, Consortium of the FP7 project EBBITS, ICT-2009-5-257852, 2011.

[8] N. Kefalakis, N. Leontiadis, J. Soldatos, D. Donsez, *Middleware Building Blocks for Architecting RFID Systems*, in Social Informatics and Telecommunications Engineering, Volume 13, Lecture Notes of the Institute for Computer Sciences, Springer-Verlag Berlin Heidelberg, 2009, pp. 325–336.

[9] K. E. Kjaer, A Survey of Context-aware Middleware, in Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering, Anaheim, CA, USA, ACTA Press, 2007, pp. 148–155.

[10] M. Knechtel, J. Hreno, F. Pramudianto, M. Ahlsen, K. Furdik, *Deliverable 4.1: Analysis of Semantic Stores and Specific ebbits Use Cases*, Project Deliverable, Consortium of the FP7 project EBBITS, ICT-2009-5-257852, 2011.

[11] P. Kostelnik et al, *Semantic Devices for Ambient Environment Middleware*, in Proceedings of EURO TrustAmi 2008, Internet of Things and Services, Sophia-Antipolis, France, 2008.

[12] P. Kostelnik, M. Sarnovsky, J. Hreno, *Ontologies in HYDRA - middleware for ambient intelligent devices*, in Ambient Intelligence Perspectives II., Vol. 5, 2009, pp. 43–46.

[13] F. M. Lardies, P. A. Rafael, J. Fernandes, W. Zhang, K. M. Hansen, P. Kool, *Deploying Pervasive Web Services over a P2P Overlay*, in WETICE '09, Proceedings of the 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, Washington, DC, USA, IEEE Computer Society, 2009, pp. 240–245.

[14] L. Lefort, C. Henson, K. Taylor, *Incubator Report*, W3C Semantic Sensor Network Incubator Group, 29 June 2011, [on line], [cit. August 3, 2011], available at http://www.w3.org/2005/Incubator/ssn/wiki/Incubator_Report.

[15] C. Pahl, *Semantic model-driven architecting of service-based software systems*, in Information and Software Technology, Volume 49 Issue 8, August, 2007, pp. 838–850.

[16] G. Santucci, *From Internet of Data to Internet of Things*, Information Society and Media, Directoriate General, European Commission, Brussels, 2009.

[17] M. Sarnovsky, P. Kostelnik, J. Hreno, P. Butka, *Device Description in HYDRA Middleware*, in Proceedings of the 2nd Workshop on Intelligent and Knowledge oriented Technologies (WIKT 2007), Kosice, Slovakia, 2007, pp. 71–74.

[18] C. F. Sorensen, M. Wu, T. Sivaharan, G. S. Blair, P. Okanda, A. Friday, H. Duran-Limon, *A context-aware middleware for applications in mobile ad hoc environments*, in Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing (MPAC 2004), New York, USA, ACM Press, 2004, pp. 107–110.

[19] M. Strohbach, M. Martin, *Towards a Platform for Pervasive Display Applications in Retail Environments*, in Pervasive Computing, Volume 10, Issue 2, IEEE Computer Society, 2011, pp. 19–27.

[20] H. Sundmaeker, P. Guillemin, P. Friess, S. Woelffl, *Vision and Challenges for Realising the Internet of Things*, CERP-IoT cluster, Information Society and Media, Directorate General, European Commission, Brussels, 2010.

[21] V. Vajda, K. Furdik, J. Glova, T. Sabol, *The EBBITS Project: An Interoperability platform for a Real-world populated Internet of Things domain*, in Proceedings of the International Conference Znalosti (Knowledge), Technical University of Ostrava, Czech Republic, 2011, pp. 317–320.

[22] W. Zhang, K. M. Hansen, *Towards Self-managed Pervasive Middleware using OWL/SWRL ontologies*, in Proceedings of the Fifth International Workshop on Modeling and Reasoning in Context (MRC 2008), Delft, The Netherlands, 2008, pp. 1-12.

Edited by: Dana Petcu and Jose Luis Vazquez-Poletti
Received: August 1, 2011
Accepted: August 31, 2011