ABSTRACT

In this paper we present LinkSmart, a generic middleware solution developed within the FP6 European Project HYDRA. LinkSmart as a middleware framework defines an abstraction layer on top of heterogeneous communication protocols. It provides services and development tools to application developers, hiding thus for them the complexity of underlying device specifics and network protocols. LinkSmart combines the use of ontologies with semantic web services, supporting thus true Ambient Intelligence (AmI) for ubiquitous networked devices. It introduces the Semantic Model Driven Architecture (SeMDA), which aims to facilitate application development and to promote semantic interoperability for services and devices. The SeMDA of LinkSmart includes a set of ontologies, and provides tools, which can be used both at application design time and runtime. The ontologies are used for both static information storage and also complex query answering purposes.

Since LinkSmart was implemented as a generic middleware, it provides a technological basis for further projects applying the middleware solution in various application domains such as energy efficiency, Internet of Things in the enterprise and crisis response.

Keywords: Middleware, Ambient Intelligence, Semantic Technology, Ontology Modelling

1. INTRODUCTION

Ambient Intelligence applications typically involve a numerous set of networked heterogeneous devices. Indeed, the speed of evolving novel technologies asks for dynamic and flexible device integration into such systems. However, the system integration processes itself, if done manually, are carried out in a rather impromptu manner, and thus lack of scalability. Further, we need to reach an unforeseeable horizon of experience with device elements (sensors, actuators, embedded devices), concerning their functionalities and proper configurations, in order to integrate them into the system. In addition, Ambient Intelligence often reflects complex system architectures making it difficult to integrate new devices and components without having to face any conflicts as side effects or further instability leading to an uncertain malicious behaviour sneaking into the overall system, regardless how careful the developers are.

An approach to overcome these issues of heterogeneity is to abstract those diverse standards in a higher layer, which offers a uniform interface. This is where a middleware comes into play. A middleware framework hides the complexity of an underlying infrastructural system technology while presenting, for instance via its API, a consistent view to application developers.
This paper describes the LinkSmart middleware, which addresses these requirements. The middleware was developed in conjunction to the HYDRA\(^1\) project, which was a 52-months European Union funded research project. LinkSmart outfits developers of Ambient Intelligence applications with a set of sophisticated software components and related development tools.

The middleware provides a transparent and secured communication channel that takes into account mobility and dynamic addressing of devices and related services. It utilizes a semantic modelling technology for representing devices or any aggregation of devices that allows application developers addressing network nodes semantically. In the following we present the LinkSmart middleware, describing its software architecture and design, device classifications, and the use of ontologies for semantic application development. We then give a brief overview of follow-up projects that apply and extend the LinkSmart middleware.

2. **LINKSMART MIDDLEWARE**

LinkSmart as a middleware framework defines an abstraction layer on top of heterogeneous communication protocols. It provides services to application developers, hiding the complexity of underlying device specifics, so that service interfaces are decoupled from the network protocol. For instance, LinkSmart utilizes OSGi Service Platform for local calls between Java-based modules and for remote calls SOAP over various network protocols such as HTTP, UDP, and Bluetooth.

2.1 **Software Architecture and Design**

Figure 1 shows the LinkSmart middleware framework and its relation to the network infrastructure and applications.

![Figure 1: LinkSmart middleware architecture](image)

Components inside the dotted square comprise the middleware. This figure clearly visualizes that LinkSmart is located between the physical communication layer and the application layer. The physical layer depicted in Figure 1 represents the different device communication protocols that can be abstracted with LinkSmart.

\(^1\) http://www.hyramiddleware.eu/
The middleware architecture follows strictly a service-oriented and component-based design adhering to the principles of loose coupling and separation of concerns. LinkSmart consists of several components called managers (see Figure 1). Each manager encapsulates a set of operations and data that realize a well-defined functionality. Some of these managers are essential (e.g. Network Manager) while others provide optional functionality (e.g. Context Manager or Storage Manager). Each manager has a clearly defined role, offering a set of services to be used by other managers or application level components. Further, as LinkSmart aims at supporting the development of distributed AmI applications, managers can be deployed on different hosts, communicating via web services. In consequence LinkSmart supports the development of scalable applications, from simply connecting two computers to full-fledged pervasive environments supporting e.g. security, distributed storage and context awareness.

The LinkSmart SoA is implemented with WS-∗ conformed Web Services based on either Java or .Net providing interoperability among different systems and platforms. Java based components make use of the OSGi⁴ service platform as it represents a comprehensive framework for the development of modular and extensible applications.

Besides such architectural considerations, LinkSmart introduces the distinction of device developers and application developers allowing developers to best apply their expertise to specific tasks in AmI application development. A device developer is responsible for connecting any kind of networked device to the LinkSmart middleware, exposing its functionalities as LinkSmart conformant services. Once integrated, the application developer can then transparently employ this device in a LinkSmart application.

2.2 Device Classification

The LinkSmart network architecture is based on IP networks with the communication scheme based on Web Service calls. If a device’s communication protocol does not implement the IP layer, it will need means to be integrated in the LinkSmart network. The way this is done depends on the device’s capability to host LinkSmart components. Therefore, LinkSmart introduces different device classes, to provide to device developers guidelines on how to integrate a certain device:

- **D0 devices** are not able to host the minimally required subset of the LinkSmart middleware and do not support IP communication. D0 devices are typically legacy devices with very limited power in terms of processor and memory using communication protocols like Bluetooth, ZigBee, IrDA or RS-232 among others. Sensors and actuators are D0 devices.
- **D1 devices** cannot host the LinkSmart middleware but do implement IP communication and are suitable for running embedded Web Services. PDAs and mobile phones are examples of D1 devices.
- **D2 devices** can host the LinkSmart middleware but do not implement IP communication. Thus, communication needs to be bridged by a device that is capable of IP. Some PDAs are examples of D2 devices.
- **D3 devices** are able to host the LinkSmart middleware and provide IP support. Examples of D3 devices are powerful mobile phones, personal computer or laptops.
- **D4 devices** are D3 devices that host proxies for D0 and D1 devices.

---

³ http://www.osgi.org/
Figure 2: Device classification decision flow chart

Figure 2 shows a decision flowchart to help developers decide how they should develop a LinkSmart device. First, we have to check whether the device can host the LinkSmart middleware or not. If the device is not powerful enough, it is a D0 or a D1 device. If the device can host a web service and has IP communication capabilities, it is a D1 device. Otherwise, it is a D0 device. On the other hand, if the device can host the middleware, we have to check if the device in question supports IP communication. If the answer is negative, we have a D2 device. If the answer is positive and the device can control D0 and D1 devices in the system, we have a D4. Otherwise, it is a D3 device.

2.3 Ontologies in LinkSmart

The HYDRA project combined the use of ontologies with semantic web services, supporting thus true Ambient Intelligence for ubiquitous networked devices. HYDRA applied the Semantic Model Driven Architecture (SeMDA), with the aim to facilitate application development and promote semantic interoperability for services and devices. The SeMDA of HYDRA, currently released as open source under the name LinkSmart, includes a set of ontologies, and provides a set of tools, which can be used both in the application design- and run-time.

The most of models used in LinkSmart are created as OWL-Lite (OWL, 2009) ontologies, in some cases OWL-DL. With respect to the characteristics of the domain, careful modelling strategy was applied. The development of ontologies strictly followed the user and application requirements to keep them simple. The ontologies in LinkSmart are used both for static information storage and for complex query answering purposes. An introduction to particular ontologies of LinkSmart is provided below. The basic structure of ontologies is depicted in Figure 3.
The LinkSmart **Device Ontology** represents concepts describing the device related information, which can be used at the design- and run-time as well. The basic ontology is composed of several partial models representing specific device information. The initial device ontology structure was extended based on the FIPA device ontology specification (FIPA, 2002) and the initial device taxonomy was adopted and extended from the AMIGO project vocabularies for device descriptions (AMIGO, 2006). The core ontology contains taxonomy of various device types and the basic device description that includes a model and manufacturer information. The additional ontology modules include the following model.

The **Device services** are modelled in the terms of operation names, inputs and outputs (Figure 4). The services are also organised into a taxonomy. The services are the basic executable and functionality units in LinkSmart. To enrich the service description, additional information items can be annotated to the model of service, such as various capabilities, quality of service or security properties etc. The model of services used in LinkSmart was inspired by the OWL-S ontology (OWL-S, 2004). Since the OWL-S was too exhaustive for the project purposes, a more suitable approach was to create simple and customised ontology for service description.
Device capabilities represent the hardware properties, software description and energy profiles. The mentioned information profiles are modelled as static structures, where only one profile of each type can be attached to the device.

Discovery models contain models of all discovery information provided by the low-level communication protocols. The Discovery model is mandatory and is attached to each device. The purpose of the device discovery information is the ability to resolve the suitable device semantic model when new device joins the LinkSmart network and is initially described only by low-level discovery information depending on the communication protocol used.

The Semantic device model represents logical aggregates of composed devices to provide a more advanced application related functionality. Semantic devices are modelled as a set of semantic services specified by preconditions, which have to be satisfied for the semantic device to be executable. The preconditions specify static or dynamic requirements for devices embedded in the semantic device.

Application models contain a set of ontologies dedicated to various application domains. Each application model specifies the domain entities and relations in order to achieve a context-awareness of the application.

Quality of Service (QoS) model contains descriptions of various aspects of the service quality. High level properties, such as taxonomy of service functional capabilities (e.g. plays video or measures temperature) are modelled. The QoS ontology contains also specification of the lower level service properties, such as response time, availability or reliability. The QoS ontology also contains taxonomy of various units (such as temperature, time, pressure, currency, etc.).

Device malfunctions represent various types of errors and failures, which may occur when using the device at run-time. For each malfunction a set of possible remedies in the form of text descriptions is assigned.

Security properties specify various security properties, such as protocols, algorithms or objectives, which may be attached to the devices or services. To describe the security properties, the third party NRL ontology (NRL, 2007) was integrated and annotated to the device ontology.

Configuration model supports a device creation using the DDK (Device Development Kit) tools. For each created device, information on the configuration and implementation files used by the particular IDE (integrated Development Environment) is stored. These files serve as templates of a code or IDE project files and can be reused, when new similar devices are created. Another purpose of the configuration models is the support for automatic device code generation (e.g. selecting suitable device implementations) for the device development.

2.4 Using Ontologies in LinkSmart

SeMDA of the LinkSmart provides a set of tools helping the application developer to use any wireless or wired device easily. All devices in the LinkSmart application are accessible in a uniform way – as a semantic web services. In order to achieve this, developer has to prepare all the devices, which will be used in the application with the help of the SeMDA tools. For each device a semantic description, which can be used for the purpose of the device discovery, calling the device services satisfying various requirements (such as a suitable quality of the service) or context-awareness of the application, is created. This functionality is ensured by SeMDA, thus the development process is simplified and the underlying implementation is transparent to the developer. This section will briefly introduce the basic scenarios of using Semantic Web technologies in application design and at the run-time.

LinkSmart enabling device

At the application design-time, each device has to be prepared for usage in the LinkSmart. This process is called the LinkSmart-enabling of the device. Developer can LinkSmart-enable a new device using so-called Device Development Kit (DDK). The new device is annotated to the suitable class in the device taxonomy (e.g. a mobile device) and the basic description, such as the device model name and number, manufacturer information, energy consumption profile or device discovery information is added. As the particular devices 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 the custom implementation. Each service is also annotated to the suitable service taxonomy class.
The whole process of the device LinkSmart enabling is guided by the ontology. The developer browses the taxonomies provided by the ontology when selecting a suitable device or service class. Basic information and energy consumption are entered into the forms, automatically generated from the ontology. Once the device is prepared, the new ontology instances are automatically generated and the ontology is extended by the new device basic model. The ontology contains one instance for each specific device model.

**Semantic device discovery**

When a new device enters the LinkSmart network, it is discovered using one of the low-level Discovery managers dedicated to various low-level communication protocols such as Bluetooth or ZigBee. In most cases, the low-level discovery retrieves only “weak” information dependent on the particular protocol capabilities. At the run-time, this information is used to identify the corresponding semantic device model in the ontology containing full description of the device, its services and other relevant information assigned to the device model at the design-time. The semantic resolution is performed by comparing the actual low-level discovery information to discovery information assigned to the device ontology templates within the LinkSmart enabling process. Each low-level communication protocol represents device discovery information in a very different way. Sometimes, the available device information includes only device model name and the number, sometimes various manufacturer information. In case of more sophisticated protocols, such as BlueTooth or UPnP, a list of services or other extending information can be available. For each low-level discovery information there exists a model in the ontology. The low-level discovery information is translated into the SPARQL (SPARQL, 2007) query and the solution of the semantic device resolution is transformed into the graph-matching problem. In many cases, the execution of the query retrieves more matching candidates, which has to be further investigated by heuristical comparison of possible additional information items. The possible additional information for each communication protocol is modelled in the ontology, so in the implementation of the comparison procedure there is no need to hard-code the particular comparison cases.

**Extending the device semantic description**

Semantic descriptions of the device models created in the LinkSmart enabling process represent only the basic information necessary for the device functionality. This information can be further extended using the Eclipse based IDE, which serves as an ontology and annotation editor. The LinkSmart ontology was extended by models of hardware, software and energy profiles, quality of service properties and security properties. The Device ontology was also extended by properties used to annotate the extended information to the device models. Since in the most cases the requirements were to search for services having several properties, the domain of annotation properties is mostly the classes from the service taxonomy. The hardware, software and energy consumption information are modelled as static structures, there can be one hardware, software or energy profile per device. There can be multiple annotations of the quality of service and security properties. Using the extended semantic descriptions, the devices and services have the full semantic support and are searchable in various ways.

**Application context-awareness**

In HYDRA, the application domain models were integrated into ontologies including properties for annotating devices to the context entities. When the developer creates the application, he or she can select, which devices will be used for the context computations. These devices can be annotated using the ontology editor IDE to the relevant context entities. Then, in the application logic implementation, it is possible to call pre-implemented and parameterised ontology search services. The parameters of the query are formulated in a specific notation developed for the purposes of simplification of the query mechanism. The query is formulated using the IDE (Figure 5), where the developer can simply select, which parameters are searched and which parameters should be retrieved for further processing. The IDE translates the required parameters into the SPARQL query, which is executed against the ontology. The search methods then retrieve all the devices matching parameters to be satisfied.
Semantic devices

Each physical LinkSmart device provides a set of specific services, which can be directly used by the application developer. The concept of the semantic device brings the idea of specifying the application specific behaviour achieved as the composition of several LinkSmart devices organised into complex units (Kostelnik et al., 2008). Simply said, the semantic devices are logical aggregates of devices. Semantic devices can include both basic (physical), but also other semantic devices. Each semantic device is defined by a set of semantic services. Each semantic service is composed by a set of requirements in terms of preconditions. The preconditions are used at the run-time to generate the candidates matching the specified requirements.

At the design-time, the developer has to define and implement semantic device services using the DDK tool. At the run-time, each time the new device enters the application, the semantic devices are rediscovered and the required devices satisfying defined preconditions are automatically tied with the semantic devices.

Implementation of the semantic device is realised as a combination of statically defined devices and the orchestration behaviour. The static definition is used only in the case, when the semantic service has to work exactly with some specific devices. But this specification does not entail any limitation for using also the orchestrated devices. For example, the developer may decide to create a specific temperature alert device using just some selected thermometers in the room, which have to be specified (thermometers are specified as the concrete devices – static mapping).

At the run-time, the presence of devices in the LinkSmart network may change. When the devices enter or leave the LinkSmart network, the ontology is continually queried and all the affected semantic devices are rediscovered. Each change may cause that some of the available semantic devices are disabled, and some may be enabled for the usage. Furthermore, semantic devices have to ensure real-time orchestration of the embedded devices. Each time when the semantic services are executed, the ontology has to infer the actually presented devices matching the specified preconditions.
3. APPLICATIONS AND PROJECTS

Adapt4EE (starting November 2011) project aims at augmenting the contemporary architectural envelope by incorporating business and occupancy related information thus providing a holistic approach to the design and evaluation of the energy performance of construction products at an early stage and prior to their realization. One of its objective is to design and implement an Open Semantic Based Middleware for Integrated Management of Multi-Sensorial Clouds, built upon the well proven Hydra middleware concepts and respective semantics, allowing for integrated, unified and consistent management of all devices comprising the multi-sensorial network as a single organization. Furthermore, an enhanced Adapt4EE Device Ontology and respective Inference Rules will be produced incorporating Energy Efficiency, Building Information (BIM) and Business Modelling (BPM) aspects. This among other issues also includes the development of an Adapt4EE ontology for combining business and asset management information with energy profile definitions. Related existing non-semantic models will be annotated, enhanced or upgraded to advanced knowledge models, like gbXML. The project starts at the beginning of November of 2011.

The BRIDGE\(^4\) project (started April 2011) aims at increasing the security and safety of European citizens through improved multi-agency coordination in large-scale emergency management. The focus is on solutions to facilitate multi-agency collaboration in large emergency relief efforts by enabling data and systems interoperability and providing a common operational picture for such agencies. The project will develop solutions for providing stable network infrastructures and interoperability among network nodes in harsh environments. Such infrastructure must be able to deal with unstable conditions i.e. highly dynamic unreliable networks, breakdown of single network nodes or whole networks. These conditions also lead to the requirement of seamless integration of various heterogeneous devices and interoperability among them. In BRIDGE, LinkSmart will be applied and extended with features of ad-hoc networking and highly dynamic network environments.

The CUBEE project, submitted proposed under CIP ICT PSP, priority „1.2 ICT for Energy Efficiency in Public Buildings“, also intends to build on the HYDRA middleware (if approved for funding). The CUBEE project is meant to set-up four pilot sites, in four different European Countries, for testing and validating an innovative ICT platform enabling the reduction of energy consumption, the reduction of up to 15% CO₂ equivalent production, improving the use of micro-generation and enhancing the comfort of students and teachers in public Campuses and Universities.

The ebbits\(^5\) project (started September 2010) envisions integration of physical devices, systems and components directly into optimised systems, i.e. managing workflows, people, processes, information and knowledge, and turn them into useful, value-added business services or service components. Its aim is to enable interoperability between various subsystems in manufacturing environments across manufacturing cells, manufacturing lines end entire manufacturing plants, regardless of geographical location with the aim to support production and energy optimization. The LinkSmart ontology models have to be expanded to cover new domains of interest. For example, the device ontology will be extended with new devices and resource consumption models for optimising energy consumption. Service composition and orchestration capabilities will be added to the services ontology and a business process ontology will be developed to enable semantic maintenance of business rules and higher-level business processes. Further, a model for events generated by devices, which may trigger specific activities in business processes will be developed.

The goal of ME3Gas (started May 2010) is to put consumers in control of their appliances to effortlessly optimize energy efficiency usage without compromising comfort or convenience. ME3Gas specifically addresses reduction in energy usage and CO₂ footprint in households. ME3Gas will use real-time energy information as energy-awareness services for all residents and combine household specific services with a community portal. ME3Gas will extend the LinkSmart middleware to develop an energy-aware middleware platform providing necessary functionality and tools to add energy efficiency features to device networks. It will build on the LinkSmart SeMDA extending existing ontologies with information about the domain and energy efficiency.

\(^4\) http://www.bridgeproject.eu/
\(^5\) http://www.ebbits-project.eu/
SEAM4US (starting October 2011) aims at developing advanced technologies for optimal and scalable control of metro stations. The project’s main outcomes will be the creation of systems for optimized integrated energy management, and the development of a decision support system to drive mid-term investments. SEAM4US will apply and further extend LinkSmart to integrate energy metering and sensor-actuator networks with existing systems (e.g. surveillance, passenger information and train scheduling), to acquire grounded user, environmental and scheduling data.

SEEMPubS (started September 2010) aims at reducing energy usage and CO2 footprint in existing public buildings and spaces without significant construction works by introducing intelligent energy consumption monitoring and control. Based on the LinkSmart Middleware, the SEEMPubS platform will provide control of appliances to effortlessly optimize energy efficiency without compromising comfort or convenience for end-users. LinkSmart will be used to integrate existing building management systems and sensor networks with new technologies to be developed within the scope of the project. An energy efficiency ontology will be developed dealing with the requirements of heterogeneous sensing and control devices and systems including building management systems (BMS), wireless sensor networks (WSN) and smart meters. The system will be installed and evaluated in a large university, bringing with it diverse domain specific requirements regarding usage of rooms, numbers of people, usage of appliances, etc.

4. CONCLUSION

We presented the LinkSmart middleware for developing AmI applications. LinkSmart solves the compatibility issues in AmI applications by abstracting heterogeneous devices and communication protocols. Introducing the Semantic Model Driven Architecture, LinkSmart provides semantic interoperability between devices and services. Furthermore, decoupling the application development from the device programming brings advantages with regard to modularity, reusability and extensibility. Application developers also benefit from this separation as they are faced only with web service interfaces instead of a broad range of communication protocols.

Besides the functionality presented in this paper, LinkSmart offers a set of extended features such as security, context awareness, quality of service, distributed storage, etc. These features help application developers build high quality applications within a short time.

Currently, LinkSmart as a middleware is being reused, applied, and further developed in a couple of European projects within different domains.

REFERENCES


http://seempubs.polito.it/