Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 69 (2014) 597 – 602

Procedia Engineering

www.elsevier.com/locate/procedia

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

Concept for a Service-Oriented Architecture in Building Automation Systems

Tobias Teich, Sebastian Wolf,Tim Neumann, Sebastian Junghans,Susan Franke*

University of Applied Sciences Zwickau, Dr. - Friedrichs - Ring 2a, Zwickau 08056, Germany

Abstract

On the consumer side exists great potential to use energy more efficiently. To exploit this potential, an intelligent control of building automation is required. The building automation market offers a variety of technologies, which do not rely on uniform standards. The programming effort for the appropriate implementation of identical functions in different locations is enormous and can be significantly reduced through a virtualization approach. In this way, real devices are represented by virtual devices that interact with stakeholders through business processes. To describe the business processes, workflows are used which consist of individual reusable and combinable services. These services were grouped into three categories. In addition to the room-/building-control-category, services are divided into metering/analysis/reporting-category and monitoring/configuration/user interface-category. The flexible combination and coordination of services must be supported. This is done by the use of a service-oriented architecture (SOA).

Selection and peer-review under responsibility of DAAAM International Vienna

Keywords: Building automation; SOA; Virtualization; OSGi

1. Introduction

The increase in building automation has several reasons. In addition to energy savings, the user also expects higher comforts, safety aspects and flexibility in the housing sector. For the implementation of the advanced requirements in building automation smart metering systems are required, which additionally feature a digital meter, internal memory and a data interface [1]. The smart building is an extension of the smart meter approach. Different home areas (air, heating, water etc.) are equipped with modern technology. Sensors and actuators are integrated into

* Corresponding author. Tel.: +49 375 536 3250; fax: +49 375 536 3104.
E-mail address: Susan.Franke@fh-zwickau.de

Selection and peer-review under responsibility of DAAAM International Vienna
doi:10.1016/j.proeng.2014.03.031

these areas to perform typical functions such as switching, setting, reporting, measuring and counting. The installed devices communicate with each other via a digital, serial data bus. Typical bus systems in building automation are KNX (European Installation Bus) and LON (Local Operating Network) [2].

2. Virtualization and OSGi

In the building automation domain, a large number of intelligent components and devices are available that cover similar applications. This diversity represents a major challenge for the development of flexible software for the control of devices. Traditionally the integrated devices and components within the housing complex communicate with each over a shared bus system and protocol. The programming effort is increased tremendously by implementing the same or similar functions at different locations up to every device. In addition, the solutions provided by a specific manufacturer are usually limited and offer only few options for optimizing the monitoring of complex processes.

By virtualization real devices are represented by virtual devices, which interact with each other and with actors by using business processes. The goals and different perspectives of the various actors affect the observation. Initially there is a main focus on the client or customer, who is defining his purposes concerning increased comfort or energy saving aspects. An optional consultant could be included e.g. for energetic aspects. Finally, the contractor is responsible for installation, implementation and realization of individual customer requirements. The flexibility of the software-architecture is achieved by the use of an OSGi framework (Open Services Gateway initiative). An OSGi framework is a dynamic Java-based application platform which supports modularization of software by use of a component model and a service registry as well as by using restrictive implementation requirements. It supports the collection of services and service implementations into bundles as well as a dynamic, individual and demand-based software configuration in a modular way. Furthermore, to develop application logic that can be flexibly used in different system environments with distinct automation technology, the service implementation has to be completely decoupled from underlying hardware devices and bus systems. Therefore, devices are described in an abstract form and can be mapped on the uniform representation. Additionally, abstract devices should not contain any functionality but rather behave like pure data structures. Any decision or interaction logic is implemented by services and business processes. Due to the decoupling, an interface between the hardware components and the description of a virtual component from the running system is necessary. Thus it is required to abstract and unify significant properties of specific device categories. Therefore device specific attributes and parameters are mapped onto generalized data structures of virtual proxy objects. This transformation process is done by hardware specific adaptors. Adaptors act as a uniform connection between hard- and software by retrieving and mapping device properties as well as transferring information from virtual proxy objects to them. Therefore the programming effort can be reduced significantly. For the virtual objects, reusable services and business processes will be developed. In addition to the services, monitoring and control algorithms will be implemented. Further, the services represent the control and monitoring processes. These services can be used for most products belonging to the same category. By the use of services, the virtual components of all communication tasks are decoupled and represent only the present condition of the real physical equipment. When required, it is possible to represent virtual equipment without a physical device. This way, services interact dynamically with the virtual equipment and use to set parameters or output results.

The scenarios which are built by the combination of single services can be developed individually or taken from a library. Given that individual scenarios have proven themselves in the application, these can be added to the service library. Besides the scenario descriptions further resources are needed for the configuration. On one hand, building information from the CAD model is incorporated, providing details on the geometry, topology and installation. On the other hand, information about the implementation which shall potentially be covered by the software is retrieved. A resource library with a catalog for equipment and services is available [3].

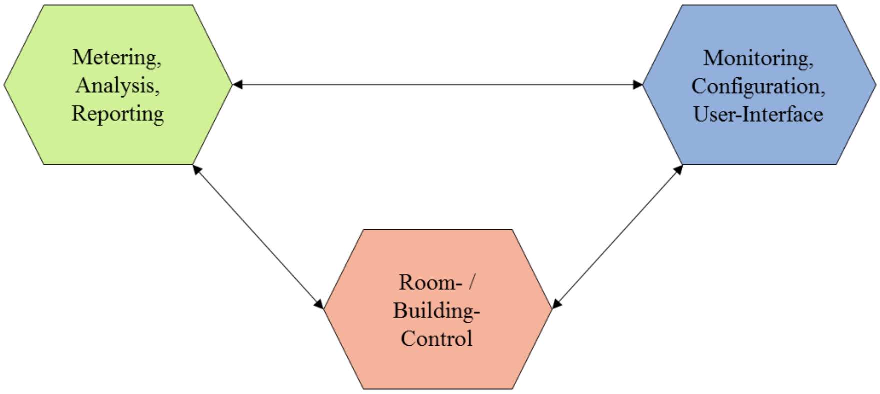

Fig. 1. Service Categories.

When using a service-oriented architecture an efficient service-oriented platform (SOP) which ensures a simple handling of the services must be implemented. In this context, the Java-based service platform OSGi is used, which provides an open architecture for the development, deployment and management of services. Modularization is the basic concept of OSGi, which aims to break down complex software into individual parts that are easier to handle. The OSGi module is a set of Java classes and resources that are packed in a Java archive. In OSGi terminology an OSGi module is called a bundle. This modularization is comparable with classic application extensions which can be linked dynamically at runtime.

From the perspective of software development, a Service Oriented Architecture(SOA) can be installed by the use of OSGi. Like services in a SOA, single software components can be implemented, updated and exchanged without problems in OSGi. Software is divided into several manageable modules and business processes into reusable services. For stable performance, a bundle corresponds to one or more services of the business layer [4].

3. Building automation services

By taking a service-oriented approach, particular sections are linked, reused and defined in a better way. In computer science, a SOA is required for the implementation of service-orientation. SOA is an IT concept that supports the execution of business processes by using comprehensive services. Services used in building automation are divided into three categories, as illustrated in Figure 1.

Primarily process services and enterprise services for room and building automation are covered. Virtual devices are connected by defined business processes that are made of such individual, self-contained services. An expansion with further service areas is possible at any time. Room and building control is concerned with services for heating, ventilation/air-conditioning, electric and water management as well as energy. An extended service to control a property is also conceivable. Figure 2 shows the structure of the service area and its respective subgroups. Besides that it shows for whom the service group is primarily designed for.

The implementation of the business processes requires system information, collected by the appropriate sensors and corresponding actuators responding to them. The service categories measurements, analysis and reporting provide data collecting tasks for other services. Services must also provide general weather and tenant data as well as utility rates for electricity, gas or water. Afterwards the data can be analyzed. That requires services for obtaining, processing and analyzing metrics.

The measured values are stored in appropriate databases or used directly as input parameters in the workflow. The tenant has to be able to access the relevant data. This can be done with a display within the home or by retrieving the information over the internet from a computer or smart phone. The provider does not only deal with the information about the individual apartments, but also with the relationships inside the building and the property. The Reporting Services must be flexible so that all relevant information can be arranged and displayed for the end user.

The third category includes the services for surveillance/monitoring, configuration and the group of user interface services that the tenants can call in order to generate additional value for themselves. Surveillance/monitoring-services values are checked for plausibility and odd variations. Additional services check if the technology used in building automation does still work correctly. If irregular events occur, the monitoring services, for example, send a message to the provider. Configuration services are used to apply changes within the apartment. A distinction can be made between configurations that need to be done only once and those who must be changed frequently. One-time configurations for example are the size of the housing units, which affect services for heating, cooling and ventilation. Frequent changes for example can occur when the desired comfortable temperature is adjusted by the consumer. The tenant initiates the user interface service, which orchestrates all other services to provide the desired performance. On one hand, the user-interface-services include added-value information for the tenant, billing and security aspects. The group other services on the other hand includes services for decreasing and increasing temperature, shadowing, ventilation, home appliances control and for controlling the presence/absence event. Figure 3 presents a possible structure of the user-interface-services.

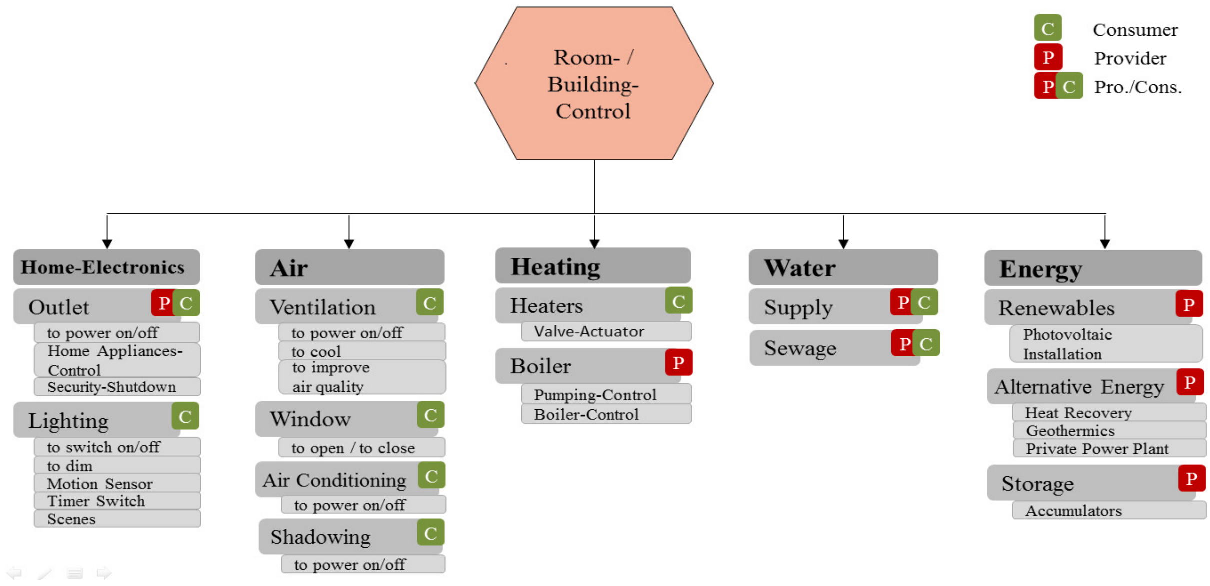

Fig. 2. Room- / Building Control Category.

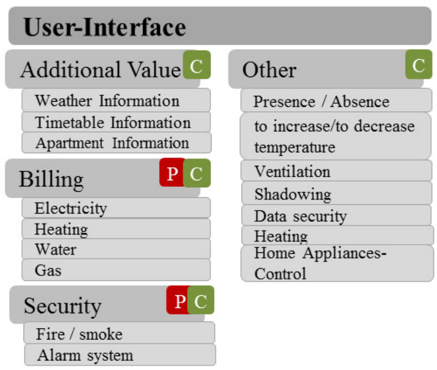

Fig. 3. User-Interface-Services.

Before explaining the presence/absence workflow, the possible components of a workflow have to be defined. The workflow structure is based on the UML notation of use-case-diagrams [5].The business process as a workflow consists of individual services. The complex work flow presence/absence works with various service areas. Temperature services are responsible for lowering the room temperature during absence. In the area of home appliances control, predefined devices (washing machine, dishwasher) are turned on during a cost-effective time. In addition, the residual current consumer services switch off all unnecessary power consumers (lights, devices in standby mode, etc.) when leaving home. All set safety aspects (windows, lighting and alarm systems) are coordinated by the security services. The interaction of individual services is described by the presence/absence-workflow. The consumer (tenant) activates the absence button when leaving. The button could be located on a touch screen-display next to the apartment door and tells the corresponding service what activities have to be executed in the home. Likewise a presence button can be pressed upon arrival to activate or reset desired settings.

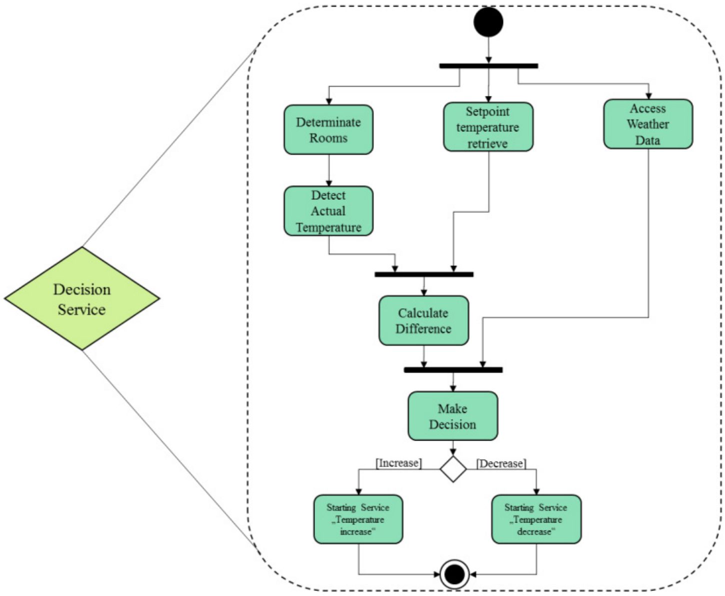

Fig. 4. Decision Service.

Each service consists of individual functions. For the representation of the structure of the individual services, the UML notation of activity diagrams is used. Figure 4shows the service structure based on the decision-making service for temperature decrease/increase of the temperature range. The consumer(tenant) informs the provider about his desired comfortable temperatures for every room. Afterwards the proper day temperatures are set accordingly. After leaving the apartment, the temperature is reduced by a few Kelvin. This process helpssaving energy.

4. Conclusion

Energy savings are made possible by more efficient controls in every area of building automation. Through the virtualization approach, devices can be implemented more quickly and easily in the building. This implementation enables a service-oriented design of business processes.

Workflows that couple the individual services are an opportunity for a structured representation of the business processes. The goal is to develop a tool, in which the provider can compose suitable business processes for the installed technology. Therefore he can access the individual services from the full service repository. If workflows have proven themselves in practice, they can be incorporated into a library for standard workflows. In addition, interfaces to connect to the implementation and building conditions from the CAD models must be integrated.

The next steps include an implementation of selected business processes by the use of work flows.Therefore required services are defined and communication interfaces are customized. Furthermore, the service-oriented architecture is extended to other areas such as AAL(Ambient Assisted Living)[6].For this purpose, services must be added to cover further aspects(activities of daily living-recognition).

References

[1] M. Schrader, K. Tschulik, Ausblick auf die Energieversorgung unter Berücksichtigungmoderner Gebäudesystemtechniken wie Feldbus KNX anhand eines Versuchsaufbaus zur Realisierung der Kommunikation mehrerer Liegenschaften über das Internet, Zwickau, 2009, pp. 38-45.

[2] H. Merz, T. Hansemann, C. Hübner, Gebäudeautomation, Carl Hanser Verlag, München, 2010, p 17.

[3] D. Kretz,T. Teich, M. Kretzschmar, T. Neumann, Development of a general virtualization approach for energy efficient building automation. International Journal ofEnergy Engineering, Vol. 3, Iss. 1 Feb. 2013.

[4] B. Weber, P. Baumgartner, O. Braun, OSGI für Praktiker,Carl Hanser Verlag, München, 2010, pp. 2-4.

[5] Object Management Group, Unified Modeling Language – Version 2.4.1, 201, http://www.omg.org/spec/UML/2.4.1/,pp. 597-616.

[6] Marschollek, Decision support at home (DS@Home) – system architecture and requirements, Research Article, http://www.biomedcentral.com/1472-6947/12/43, pp. 5-6.