Generic Middleware for User-friendly Control Systems in Home and Building Automation

Armin Veichtlbauer
Josef Ressel Center for User-Centric
Smart Grid Privacy, Security and Control
Salzburg University of Applied Sciences
Puch/Salzburg, Austria
armin.veichtlbauer@en-trust.at

Thomas Pfeiffenberger
Advanced Networking Center
Competence Field e-Tourism
Salzburg Research Forschungsgesellschaft
Salzburg, Austria
thomas.pfeiffenberger@salzburgresearch.at

***Abstract*—In the field of Home Automation and Building Automation systems, the lack of interoperability of subsystems constitutes a major problem, especially for the integration of subsystems of different vendors. In order to overcome this drawback, our research group developed a concept of a generic control framework, which allows for integration of heterogeneous subsystems in an easy to control manner. This control framework contains functions to provide a dependable and secure control system for various Home Automation respectively Building Automation applications. To achieve that, the framework must be able to handle multiple users with different access rights using a variety of applications, as well as multiple devices (sensors, actuators, controllers, PCs, switches, routers, etc.) with different algorithmic roles. As a proof of concept, selected functions of this framework have been implemented and tested at a local test site. In this paper, we outline the architecture of the framework, describe the centerpiece of this architecture (i.e., the middleware layer), and show some results of the validation process.**

***Keywords*—Home Automation; Communication Infrastructure; User Control; Generic Interfaces**

I. INTRODUCTION

As stated in [1], Home Automation (HA) and Building Automation (BA) systems usually consist of a variety of different sensors and actuators (field level / field zone) as well as control devices (automation level / station zone), which are interconnected via several field bus technologies, like European Installation Bus (EIB), Modbus, Local Operating Network (LON), Digital Addressable Lighting Interface (DALI), etc. Alternatively, radio or powerline communication may be used to reduce mounting costs, especially for already existing surroundings. The management level / operation zone, if existing, supervises and controls the automation tasks; in many cases this is realized via web-based services in order to allow a remote control of the automation applications, possibly using smartphones [2].

The market for HA and BA solutions has been rapidly growing in recent years; yet in most cases buildings are not equipped with an integrative solution from a system provider, but with individual solutions for different building automation applications [3]. The lack of interoperability of these heterogeneous solutions prevents the shared use of existing equipment, e.g., information from access control systems (like the number of persons in certain parts of a building) could be a valuable input for evacuation support systems in cases of danger, but is usually not accessible due to the proprietary nature of both solutions. Especially for home users, which do not aim to afford an industrial sized solution for HA, this situation is very unsatisfactory, as the management of distinct island solution is not only a cost factor, but also uncomfortable - both the costs and the lack of user friendliness have been identified as big market barriers for HA [4].

A. Research Goals

Our approach to overcome the mentioned drawbacks was to define a framework, which uses open protocols and generic standards at every communication layer according to the OSI reference model [6] and at every level of the automation pyramid. Thus, every control application supporting these standards can use the functionality of our HA/BA framework without the need for individual adaptations. We conducted a thorough requirements analysis to determine the functions, which had to be added to these underlying technologies in order to form a working solution. Based upon this analysis we derived our architectural model, which we referred to as "X-Model", consisting of infrastructure, middleware, and application layer respectively. The middleware layer was designed as a convergence layer on All-IP [7] basis, which allowed us for keeping the framework architecture simple, while facilitating the integration of several applications of different vendors as well as the use of different network infrastructures.

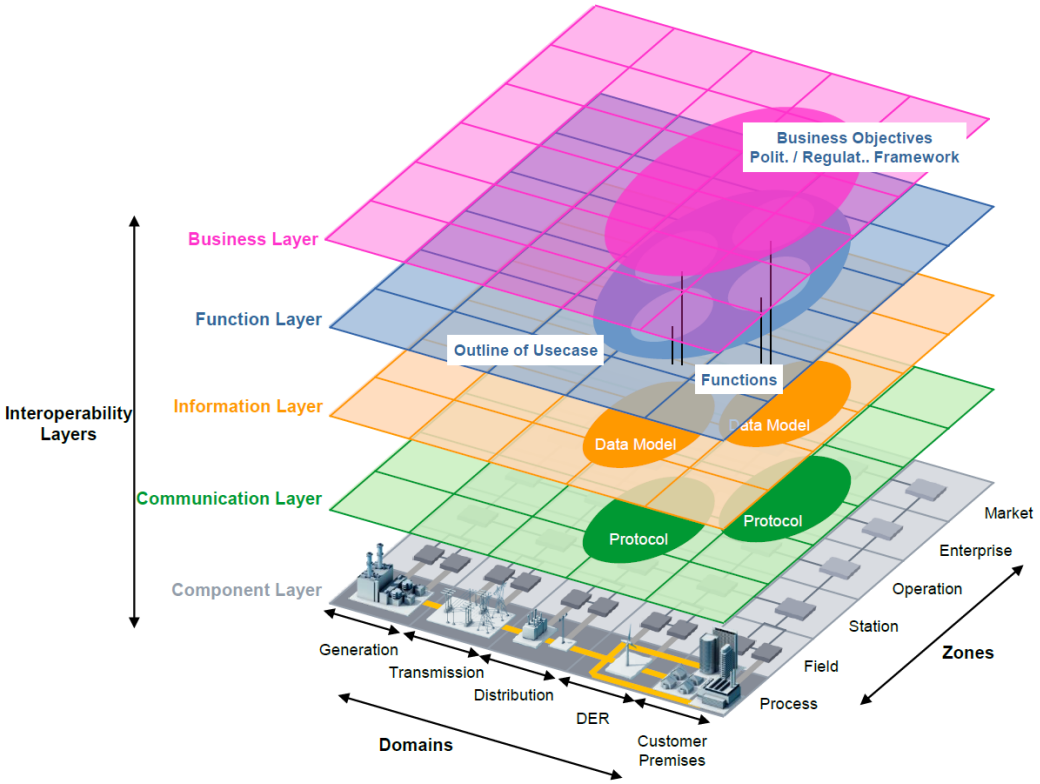

Figure 1. M/490 Smart Grid Architecture Model [5]

In the already finished research project "ROFCO" (Robust Facility Communication) [8], which was funded by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT), we developed the generic control architecture for use in a HA/BA surrounding. We implemented selected middleware functions and tested them using applications like lighting control and blinds control [1]. Hereby, the implementation of these applications as well as the setup of the testbed infrastructure have been performed for validation purposes only; conceptually, these parts formed the test environment for the actual proof of concept, i.e., the middleware.

During the current work in the "Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control" (also referred to as "EnTrust") [9], which is funded by the Austrian Federal Ministry of Economy, Family and Youth (BMWFJ), we use this architecture to deploy Smart Grid applications like demand response or energy monitoring as well as for health monitoring in an Ambient Assisted Living (AAL) environment. Obviously, Smart Grid applications induce additional requirements compared to stand alone HA/BA systems, especially regarding security and privacy, since data is exchanged with external parties like utilities.

The architectural concept of the X-Model, however, seems suitable also for this extended functionality [10]. In terms of the M/490 standardization mandate [11] of the European Commission for the Smart Grid area, our approach complies with the customer premises domain (i.e., HA) of the M/490 Smart Grid Architecture Model (SGAM) [5], as shown in Figure 1. This current work of our research group is to be published in further follow up papers.

B. *Scope of Paper*

In this paper, we will give details about the architectural framework depicted as X-Model with a special focus on the core functionalities, i.e, the middleware layer. This layer contains functions to provide sufficient dependability [12], especially for highly safety relevant applications like evacuation guidance in case of emergencies. This paper extends our conference paper [1], where we presented the basic points of our architecture as well as some validation issues, by providing additional information about the theoretical background of this architecture, i.e., the requirements, design principles, and the specification of our solution. Hereby, we will pay special attention to the middleware layer of our architecture, as in our X-Model approach

this is the core part containing the business logic; conceptionally, this part has to be able to work with the whole plethora of possible applications on the higher abstraction layer, as well as with all common HA/BA infrastructures on the lower abstraction layer.

We will start giving an overview about the work of other research groups in that area, and will also investigate the state of the art in industrial solutions in HA/BA. We will then work out the functional and non-functional requirements for a generic control solution, resulting in our architectural approach. We describe the three layered architecture we propose for a generic HA/BA control framework (X-Model). Then, we give a brief description of the testbed infrastructure we used for validating the middleware functionality including the specification of network parameters and participating devices and HA appliances. After that, we will give a detailed description of the business logic in our middleware layer, containing the core functions of our X-Model architecture to ensure dependability in our framework. Here, we define the roles, which have to be implemented by the participating devices, and make an assessment of several potential solutions we could use to fulfil the ascertained middleware requirements. This is followed by some implementation issues and a short overview of the tests we conducted at our testbed in order to validate our approach. We conclude with an outlook and some open research questions for future work.

II. Related Work

The heterogeneity of HA/BA solutions has been identified as a potential barrier for HA/BA technologies since about the turn of the millennium [13] [14]. Big vendors may offer integrative solutions, e.g., "Total Building Solutions" from Siemens [15] or "Raumtalk" from ABB [16], yet based on proprietary communication and control technologies. Several research teams have tried to overcome this barrier by proposing interoperability features for HA/BA systems, e.g., via gateways between field bus technologies [13], or by providing complete HA/BA architectures for interoperable HA/BA applications [2] [17]. For communication infrastructures, the idea of using the IP standard is not new [14].

A fully integrated approach, however, requires solutions for the whole automation pyramid, i.e., on every level of the control process: setting and getting values at field level, performing a control task at automation level, and supervising this at management level. A standardised middleware for that purpose needs to provide more than just IP communication; especially, a generic modelling of BA objects and variables is inevitable. For that purpose, the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) defined the Building Automation and Control Networks (BACnet) standard [18]. With BACnet, complete HA/BA environments could be built based on one generic technology [19]; yet in reality this approach has several drawbacks:

- The computational power required by the BACnet protocol suite is rather high, thus many field layer devices are not able to implement the BACnet stack, i.e., these devices have to be integrated via gateways.
- The support of the very common IP protocol is weak, as it is not part of the native BACnet stack. A work around named BACnet-IP is provided, i.e., a tunneling of BACnet messages through an IP network.
- State-of-the-art network management concepts like Quality of Service (QoS) are not supported with BACnet, which is especially critical with the use of safety or security relevant control applications [20], as they require very high dependability standards, especially concerning availability of communication infrastructure.

The definition of the Object Linking and Embedding (OLE) for Process Control - Unified Architecture (OPC-UA) standard [21], which is already commonly used for the control of industrial production [22], may help to overcome these shortages. OPC-UA is an interoperability standard originally based on Microsoft's Distributed Component Object Model (DCOM) standard, which facilitates reading and writing access to distributed field components (OPC Servers), which can be used by industrial Supervisory Control and Data Acquisition (SCADA) applications (OPC Clients) for their respective control tasks. By using OPC-UA in combination with TCP [23] as transport protocol it is possible to integrate IP networks and all the QoS mechanisms existing for the TCP/IP protocol stack. Some academic implementations of OPC-UA for HA/BA systems are already existing, e.g., the solutions of the TU Vienna [24]. Yet the requirements for end systems still are rather high, resulting in the necessity to provide gateways to legacy systems containing older devices with not sufficient computational power.

In BA, the use of industrial SCADA systems, which contain drivers for many different BA solutions, is a feasible approach and thus offered by BA vendors, e.g., [25] [26]. As a consequence, a suitable device for the management level (capable of running the SCADA software) has to be used, i.e., in most cases a device having the same computational power as a PC. This seems no problem for BA; for HA, however, such a supervising device at management level embodies a barrier for spreading the market widely - for HA, smaller, cheaper

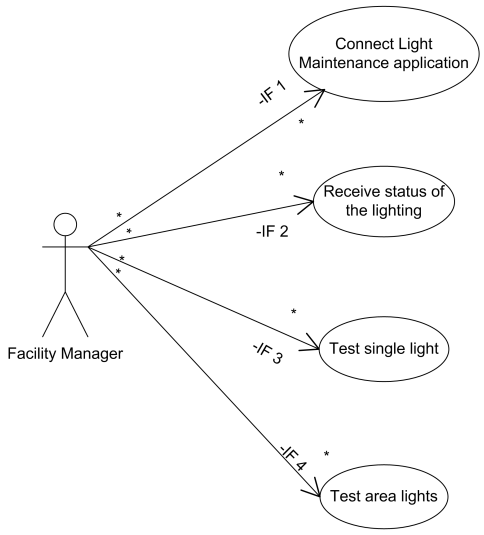

Figure 2. ROFCO Use Case Light Maintenance

and easier to deploy solutions have to be found, i.e., lightweight SCADA systems that can be addressed as web services or work as smartphone apps. Such systems are sometimes referred to as "Mini-SCADA" and are offered to end-users of HA/BA systems, but partially also to other stakeholders like energy utilities [27].

As pointed out in [28], interoperability issues are still an unsolved problem in HA, and constitute thus an important market barrier for HA solutions. For the Smart Grid area, the need for standardization has been clearly identified, e.g., in [29], but from the point of a HA customer, a smart building contains a variety of applications, which have to be included in a trusted user domain [30]. In that context, safety and security topics are of notable interest in order to produce saleable solutions [31], as open systems are always prone to outages [32] in consequence of improper use or even planned attacks.

There are some further research activities in the area of HA/BA systems. These include topics as control strategies and technologies [33], as well as performance issues [34]. Besides the technical research fields there are multiple socio-economic research activities, focussing on the potential impacts of the studied technologies on end-users.

III. REQUIREMENTS

During the requirements engineering process, we identified user stories in cooperation with the ROFCO project partners, especially with the Techno-Z Salzburg, which hosted the testbed for the validation of our approach. Hereby we were considering the interests of different stakeholders, e.g., fire fighters, public authorities, or end users. We then extracted the respective use cases from the user stories and depicted them in the Unified Modelling Language (UML), as shown in Figure 2. From the explored use cases we derived the general requirements, which we then broke down to concrete technical requirements.

A. Requirements Analysis

The challenge of the requirements analysis for our intended generic dependable HA/BA solution, which we called the "Dependable HA/BA Framework" (DHF), was to support the different and complex requirements of a variety of heterogeneous HA/BA applications. Conceptually, all thinkable HA/BA applications must be included in order to provide the required genericity. Yet as the requirements engineering process was based on use cases, we had to choose applications controlling typical HA/BA appliances, but not too similar and thus providing an as complete range of requirements as possible. At the end, we decided to base the requirement analysis of the DHF on three potential HA/BA applications:

- Lighting Control
- Blinds Control
- Evacuation Support

The first two applications also built the basis for our validation process (see Section VIII); the last application, however, was important for the requirements analysis in order to assess additional non-functional (quality) requirements, especially regarding safety and reliability [35]. As mentioned, the use of our architecture in the Smart Grid area creates further requirements. These are currently explored and thus not part of the original requirements engineering process described here.

In the following, we describe the requirements engineering process based on the exemplary application Lighting Control. First, the Lighting Control user story was defined in cooperation with the Techno-Z Salzburg as mentioned above. Since different user types (stakeholders) are involved, the user story contains different roles and activities based on appropriate authorization mechanisms. Roles define the rights to perform simple atomic activities, like receiving or sending messages from a user interface to some control units, sensors, or actuators in the DHF. Thereto the different components must support authentication, authorization, and encryption. To integrate already installed systems to the DHF, mediators are used to adopt and translate the respective messages. For this user story, we derived appropriate use cases by grouping atomic activities to expedient units. The resulting use cases cover not only direct lighting control in the building (on/off or dimming of certain

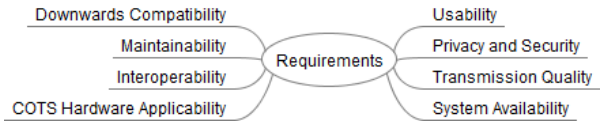

Figure 3. General Requirements

lights), but also procedures for evacuation situations, maintenance, holidays, alarms or personalized control procedures (e.g., on/off or dimming of a user-defined group of lights). For instance, the use case Light Maintenance depicted in Figure 2 consists of the following atomic procedures:

- Connect the Light Maintenance application to the DHF.
- Receive the status of the lighting (on/off - dimming status - failure) in the configured area.
- Set a single light (on/off - dimming) and check the status.
- Set a group of lights (on/off - dimming) and check the status.

B. General Requirements

After having defined the use cases for the aforementioned applications, we derived the general requirements on our DHF, as depicted in Figure 3.

- *Downwards Compatibiliy*
  First of all, the support of legacy systems must be guaranteed, as the acceptance and the price of new systems built from the scratch would prevent an economic exploitation of the solution. This holds simply for the fact, that existing parts of HA/BA systems have to be reused to keep the costs as low as possible, and that users might tend to use solutions they already know.
- *Maintainability*
  The whole system has to be easily maintainable and configurable. Most important, the integration of new devices must be working in a plug and play manner as far as possible. Clear enough, by having a rights management concept [36] limiting the use of devices, applications and data to users with respective rights, some configuration tasks will be unavoidable. All necessary configurations have to be performed in a user-friendly way, and supported by suitable tools, like wizards, as far as possible. As the degree of automation shall be adjustable, this may include decision support systems. For instance, when including a new sensor, the rights management system could provide suggestions about the users' rights by assessing the existing rights of similar sensors.
- *Interoperability*
  One of the most crucial requirements is interoperability, i.e., devices from different vendors must be integrated seamlessly to guarantee an easy access to the whole functionality for the respective users. This is ensured by the use of standards and open protocols, most important by the use of the IP as basic network layer protocol. Proprietary solutions should not be used as far as possible, and if it is unavoidable due to a lack of open solutions, the interfaces to these proprietary parts have to be defined clearly. Some proprietary solutions provide at least open application programming interfaces (APIs), on top of which our functionality could be realized.
- *Applicability of COTS Hardware*
  A main requirement of our system is to use commercial off-the-shelf (COTS) hardware. As the hardware must support high reliability and calculable availability, the mean time between failure (MTBF) and the mean time to repair (MTTR) metrics of each hardware device must be known in order to derive the system's overall availability. For authentication and authorisation well established mechanisms have to be used, such as ITU-T X.501 [37] or IETF Radius/Diameter [38] [39]. Encryption is a further main requirement to establish a secure connection over a distributed heterogeneous communication system. For the underlying network functionalities, classical network devices like Cisco switches and routers [40] are used. Address management and routing are based on IP, routing metrics [41] must be supported.
- *Usability and User-friendliness*
  A basic quality requirement of our middleware is to provide means to control several appliances (e.g., electric lighting) for different types of stakeholders (e.g., end users, home owners, etc.). This includes freedom of choice for using more or less automation: For instance, user A might want to have a fully automatic control of room temperature, which is configured once and then working continuously, whereas user B wants to manually control the room temperatures in order to have a greater flexibility. Although there are no commonly accepted metrics for user-friendliness, the integration of customer choice mechanisms in HA/BA seems indispensable in order to raise user acceptance [42].

- *Privacy and Security*
  The use of open systems, which are accessible via Internet to enhance user-friendliness, has some drawbacks concerning privacy and security. As it is not possible anymore to build closed ecosystems, which are per definition not accessible to potential fraud, we have to face unexpected and unauthorized use of system resources, up to the possibility of attacks, e.g., denial of service attacks damaging safety functions, or intrusions to get access to private data. This is especially risky for distributed systems, e.g., energy sharing communities in settlements. For instance, the exact knowledge of energy consumption of a household could be used to identify the currently watched TV program [43]. Thus, a complete authentication, authorization, and accounting (AAA) system in connection with a suitable encryption technology is necessary to enable authorized access only. Furthermore, countermeasures against potential attackers and methods of ensuring the privacy of data (e.g., data aggregation) have to be considered.

- *Data Transmission Quality*
  An overall requirement in a dependable infrastructure is to guarantee the transmission capacity and the transmission quality. Thereto some Quality of Service mechanisms in the communication infrastructure are required, such that the different network components and applications are able to label the data packets according to the transmission quality requirements. Luckily, IP supports the labelling of the packet by using the so called "Type of Service" field [7].

- *System Availability and Reliability*
  Last but not least the required dependability [12] of the intended solution has to be guaranteed, in terms of availability and reliability [44]. The availability can be assured by a process life cycle management according to [45], defining availability metrics dependent on applications' risk parameters like probability, avoidance possibility, frequency and consequences. Reliability is issued by several testing methods; for the validation of our prototype we used functional tests of the implemented components, yet this was not the core of our research, as the realized prototype works basically as proof of concept. Thus, for validation of commercially saleable solutions a much more exhaustive testing process would be required in order to facilitate the keeping of existing standards and regulations (see Section VIII).

C. Technical Requirements

From these high level requirements we derived concrete (functional and non-functional) technical requirements for the DHF. The non-functional requirements basically concern the quality of the underlying communication infrastructure, which we take as given in order to be compatible to existing solutions. This quality is assessed in terms of:

- Bit Error Rate (BER)
- Redundand Networkpaths
- Attack Robustness
- Catastrophe Robustness
- Data Packet Prioritization
- Deterministic Delay Bounds
- Network Size (number of end devices)
- Data Rate
- Range (Link length)

The functional requirements concern the necessary functionality of the DHF for users in order to perform their monitoring and control tasks in a secure manner. Thereto a rights management is indispensable, as different users (and user types) may share access to the same appliances. Thus we have derived the following functional requirements:

- Sensor/Actuator Interaction: Means to collect sensor data and to apply control strategies to actuators
- Data Structure and Representation: Means to represent, store, and query data used to control several appliances in a HA/BA environment
- Signing and Encryption: Means to label data and to avoid unauthorized use thereof
- Authentication and Authorization: Means to enable the identification of users with respective access rights
- Registration and Discovery: Means to manage devices, applications, and users combined with automated detection of changes
- Notification and Alarming: Means to notify users in case of the fulfilling of defined conditions and to throw alarms in case of unexpected conditions like limit violations
- Abstract Address Scheme: Means to identify and address devices in a unique manner
- Heartbeat / Keepalive: Means to check whether crucial system parts are up and running

IV. ARCHITECTURE

After having finished the requirements engineering process, the resulting technical requirements for dependable generic HA/BA systems could be grouped in two layers: infrastructure requirements and middleware requirements [46]. This resulted in a layered approach, where the infrastructure functionality can be separated from the middleware functionality and the application themselves, which use the middleware and infrastructure functions.

Moreover, as our goal was to integrate different applications as well as different infrastructures, this would result in a N:M relationship in case that each application would have to run on each infrastructure. In order to avoid that, we had to introduce a convergence layer in the core of our architecture, thus forming what we called the X-Model.

Basically, this is a three layered approach as shown in Figure 4, where the middle layer serves as convergence layer, which can be used by all considered HA/BA applications, and which uses several considered infrastructure technologies (i.e., those that are suitable to meet the infrastructure requirements as defined in the requirements analysis):

- An infrastructure layer (INF), which embodies all the necessary networking functionalities and end devices for our control architecture
- A middleware layer (MID), which provides appropriate dependability [44] [47] means on an end-to-end basis
- An application layer (APP), which is responsible for the distributed control tasks of the applications using our architecture

A. Infrastructure Layer

As for the network infrastructure, we intended to use an All-IP solution, which is “Layer 2 agnostic”, i.e., that is able to run on a variety of lower layer technologies, including those field bus systems, which are common in the area of HA/BA. By this strategy it was possible to natively integrate numerous devices, as long as they are able to speak IP and are able to deploy the dependability functionality of our middleware. SCADA systems, e.g., “Zenon” from our project partner Copa-Data [48], can thus be integrated by providing an open software interface containing IP sockets. Due to this openness several SCADA manufacturers may share different end devices and data servers; i.e., our solution provides a holistic concept to integrate global dependability means, opposed to currently available island solutions. Thus, a “dependability domain” is generated, which is realized by our DHF.

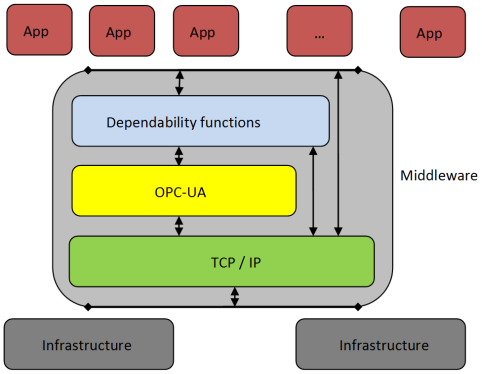

Figure 4. Generic Architecture (X-Model)

However, the integration of legacy components, which are not IP capable, could be done only via gateways, as shown in Figure 5. At this place, information loss can not be avoided completely, as the legacy devices do not necessarily support all required parameters. As a consequence, the guarantees for dependability can be made only for the natively integrated components. In spite of this drawback, the use of legacy components may enrich the dependability domain, e.g., by the integration of additional sensors - yet these components are not an integral part of the dependability domain. In this case, the parameter mapping has to be defined at the respective gateway, which is then providing these data in a dependable manner for all system integrated applications, thus providing added value. The other direction, i.e., the control of actuators outside the dependability domain, is also possible in principle, yet the dependability properties can then be mapped only partially, depending on the mechanisms of the legacy components. In both cases, the scope of the dependability domain ends in the gateways.

B. Middleware Layer

The main goal of the generic architecture was to ensure dependability [44], i.e., robustness, reliability, availability, maintainability, safety and security. For instance, by ensuring interoperability in the way that applications should have access to the whole network and sensor/actuator infrastructure, the danger of potential misuse arises; this implicates the necessity to define appropriate security means in order to avoid damages. Safety relevant applications require high standards of reliability, availability and robustness. Thus, the core functionality of the middleware layer was to provide appropriate means to facilitate and document the fulfillment of these dependability requirements within the dependability domain, i.e., the scope of the control architecture consisting of natively integrated and fully functional devices.

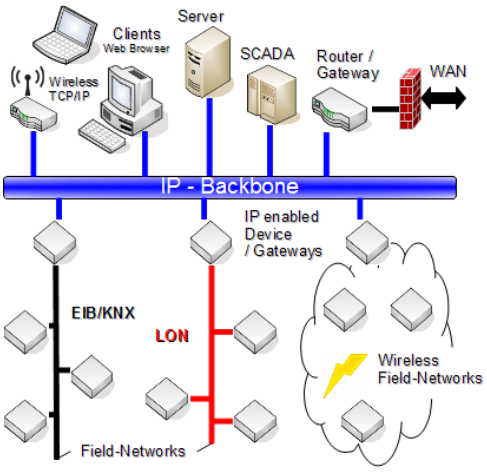

Figure 5. Generic Network Infrastructure

C. Application Layer

The application layer comprised several control logics, e.g., implemented by a Programmable Logic Controller (PLC) or a Direct Digital Control (DDC), at automation level as well as supervisory tools for end users at the management level (SCADA / Mini-SCADA), both functionalities based on working middleware implementations. Consequently, the control logics should run on devices, which are capable to host the complete middleware, as otherwise the dependability can not be fully ensured. Legacy controllers could be operated in a way, that they provide information to the dependability domain (which can for instance be evaluated and visualized by a SCADA system), but were not an integral part of the dependability domain.

According to the layered approach, following issues had to be done in parallel after finishing the design phase:

- First, we had to specify a network / hardware architecture, which was able to meet the identified infrastructure requirements.
- Second, we had to define the middleware functionality and to determine, which functions thereof we wanted to implement with our framework prototype.

These questions are addressed in the next two sections; this is followed by some implementation issues, as well as a description of the validation process and its results. The validation process comprises the setup of a real-world testbed according to the infrastructure specification, the conduction of necessary functional tests with the implemented prototype, as well as an evaluation of results.

V. INFRASTRUCTURE

As our framework should work with all multi-vendor infrastructures fulfilling our requirements, our aim was not to implement yet another infrastructure technology, but to choose suitable existing solutions. Thus, the functionalities of potential infrastructure technologies, e.g., providing appropriate link layer mechanisms, have been assumed as given. For validation purposes we had to set up a testbed infrastructure suitable to provide all required mechanisms for testing our proof of concept implementation (test environment); yet this actual proof of concept contained only middleware functions (system under test).

Basically, infrastructure technologies consist of two parts: the participating devices (which we intended to use as they are in order to ensure optimal compatibility with existing HA/BA solutions), and the lower layer network functionality (which is specified within the OSI reference model [6]).

A. Testbed Network

With given functional properties, we still had to assess the potential communication infrastructure technologies regarding their non-functional properties, i.e., the fulfilling of quality requirements, before setting up the testbed network physically. We had identified four potential infrastructure technologies, which could be used as a basis for the testbed we intended to set up at the test site of our ROFCO project partner Techno-Z Salzburg: Ethernet, Wi-Fi, ZigBee and Powerline.

Table I shows the matching of the quality requirements for these communication infrastructure technologies. The mentioned All-IP approach of our architecture guarantees the required Layer 2 agnosticism by definition [7]; furthermore IP is a protocol that had proved its ability to work in generic network systems for decades (and thereby functioning with a variety of different PHY and MAC layer protocols according to the OSI reference model [6]). Thus, it was a quite logic decision to use an All-IP approach for our HA/BA architecture. As a consequence, we could choose the concrete Layer 2 technology freely, provided that the chosen technologies meet our above defined requirements.

As Ethernet provides good quality regarding the BER metric, as well as convincing scalability properties, we decided to use it as base technology, extended with a WiFi access point in order to provide the required redundancy. Additionally we installed fiber channels to connect the different buildings of the test site. The usage of this combination of communication technologies as network infrastructure for our testbed kept the installation effort low, as Ethernet cabling was already present in all buildings of the test site.

Table I. INFRASTRUCTURE REQUIREMENTS

	Ether net	IEEE 802.11	Zig Bee	Power line
Bit Error Rate (BER)	++	– –	–	+
Redundand Networkpaths	+	++	–	– –
Attack Robustness	+	– –	– –	++
Catastrophe Robustness	– –	+	+	–
Data Packet Prioritization	++	++	+	–
Deterministic Delay Bounds	+	– –	+	–
Network Size [# end devices]	2^{48}	2^{48}	64k	2-50
Data Rate [Mbit/s]	10-1000	11-54(600)	0.02-0.25	10-200
Range [m]	100	1-100	1-100	200-300

Figure 6 shows the network topology of the testbed, which expanded over three buildings (3, 10, 12) at the Techno-Z. It was basically composed of two class C IP subnets:

- The management subnet of the Techno-Z used in Building 10 and 12
- The control subnet from the ROFCO laboratory at Building 3

In both subnets we used switches with two redundant GBIC ports, thus connecting both subnets with redundant fiber connections between Building 3 and Building 10. A third switch in the ROFCO laboratory built the interface to the various ROFCO servers. As part of the robustness concept these (manageable) switches were configured with the spanning tree (STP) mechanism. Due to the security concept two Virtual Local Area Networks (VLAN) I and II were configured on these three main switches, i.e., the devices connected to these switches could be run in both VLANs.

Both subnets were connected with respective company networks (Techno-Z and Salzburg Research) via a router/firewall combination. For further security issues an internal sniffer was installed to monitor the traffic inside the control and management subnets. Both functionalities, along with an intrusion detection system (IDS), could be performed by using the "MF-Security-Gateway" [49] from the ROFCO project partner Underground8.

B. Testbed Components

Besides defining the network parts of our infrastructure, we had to address the question of end devices. Whereas we had been free in the choice of network components (only provided that they meet our requirements), we had to use existing devices for the respective control tasks we wanted to perform in the validation of our prototype, since the project's system context (and thus the applications we used within this context) was defined by the Techno-Z as host of our testbed. As technology park the Techno-Z expressed its project interests in very concrete facility management tasks, which we formulated as UML Use Cases during the requirements analysis. Each building at the Techno-Z is equipped with different BA systems, e.g., a Somfy system to control blinds and a Sauter system to control the lighting and all heating, ventilation, and air conditioning (HVAC) components via EIB/KNX. In the following, we describe those components, which we have researched as part of the heterogeneous ROFCO testbed, grouped to their location.

- *Somfy Control, Building 10*
  To control the blinds of the Buildings 10 to 15, the Somfy blind control was separated into three zones. In zone one, a single Somfy control system at the 3rd Floor regulated the whole blinds for Building 10. At this place a controller of our project partner cTrixs called "cTrixs Base Unit" (CBU) [50] was installed, which served as gateway between the blind circuit (over relay control and digital I/Os) and the Ethernet wiring, which offered the connection to the switch in the ground floor.
- *Facility Management Room, Building 12*
  For managing the BA systems for the Techno-Z complex, a control computer was situated in the facility management room in Building 12 on the ground floor. On this computer the Sauter BA system (which includes the HVAC capabilities) or the Designa access control systems were visualized. Also the fire alarm center was located in this room.
- *Engineering Room, Building 12*
  The Sauter BA system, the EIB lighting system and the central switch were located in the engineering room at the ground floor in Building 12. The entire building is wired from this switch. For the ROFCO network a port on the central switch was reserved and activated. There was also the possibility to configure VLANs on this Catalyst 2950 switch. A second cTrixs controller provided the interface to the EIB lighting in the congress room in Building 12; it was connected to the central switch and to the EIB bus to control the lights at the ground floor.
- *ROFCO Laboratory, Building 3*
  The laboratory was equipped with a cTrixs Application Server (CAPS) and a Zenon Server from Copa-Data with master/backup function.

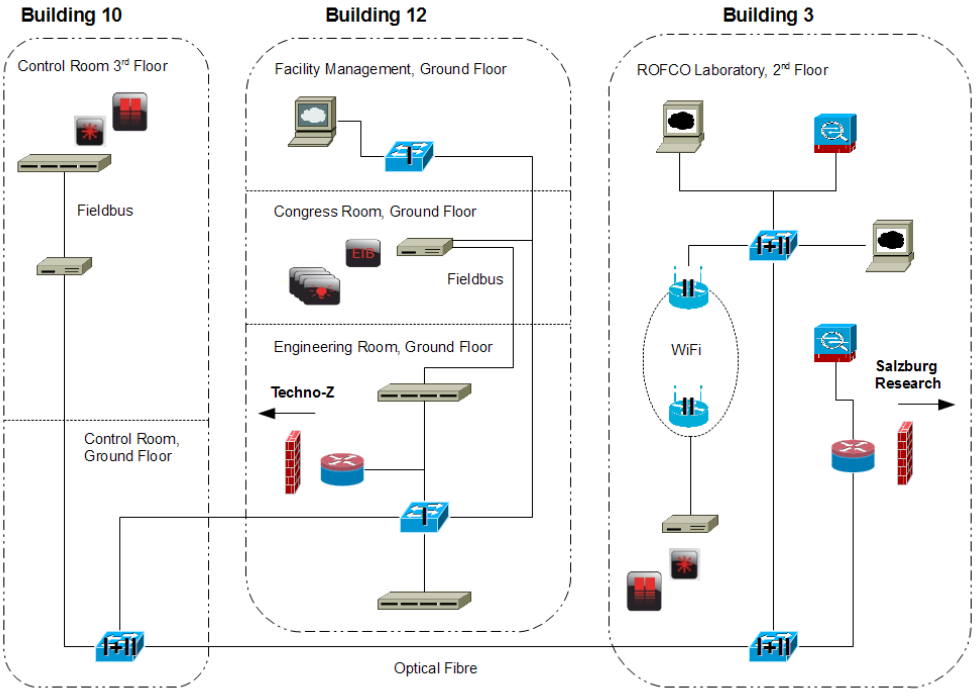

Figure 6. Testbed at Techno-Z Salzburg

With the Zenon SCADA software the use cases we considered in ROFCO could be visualised and controlled. The CAPS was used as a central server for the cTrixs controllers.

At the ground floor in Building 12, the lighting was not fully represented in the Techno-Z's building management; the same applied to some blinds control functions (e.g., open all blinds at one side of the building simulateously). Thus, the respective data points and functions were implemented and visualised on the CAPS and Zenon surfaces and controlled via cTrixs controllers. In Building 3, the blinds were handled by an IP-enabled cTrixs controller, but in opposition to the solution in Building 10, the connection was done directly via analog outputs and relays, and not via EIB. A Wireless Local Area Network (WLAN) bridge has been installed to transmit data to the controller.

VI. MIDDLEWARE

According to the outcome of the requirements analysis and the architecture design process, the middleware layer has to provide means to establish a dependable end-to-end communication between different entities, thus supporting independent distribution of control information between different end systems. This includes not only availability and safety of end-to-end communication, but also an information security and rights management concept [36] [32]. Furthermore, the middleware layer comprises added value: generic data structures (e.g., SensorML), supervising functions, etc. These concepts are detailed in the following.

The middleware layer can make use of the underlying infrastructure layer, which is guaranteeing for the meeting of the lower layer requirements, i.e., requirements for devices and communication links between them. In opposition to that, the middleware layer addresses end-to-end concerns only. It is feasible to address some properties at both layers: For instance, link layer security measures may prevent unauthorized listening on the channel, whereas transport layer security provides end-to-end encryption and authentication to prevent man-in-the-middle attacks. This may be redundant, but relying on link layer security measures is risky, as one unsecured link would jeopardize the whole security concept.

A. Survey of Base Technologies

The targeted functionality is addressed by a number of existing technologies, from commercial products to open protocol standards. Therefore, a new implemen-

Table II. MIDDLEWARE REQUIREMENTS

	OPC -UA	Modbus -TCP	SIP	Soap WSSec
Sensor/Actuator Interaction	y	y	n	n
Data Structure & Representation	y	n	n	n
Signing and Encryption	y	n	y	y
Authentication & Authorization	y	n	y	y
Registration & Discovery	y	n	y	y
Notification & Alarming	y	n	y	y
Abstract Address Scheme	y	n	y	y
Heartbeat / Keepalive	y	n	n	y
Further Robustness Features	y	y	n	n

tation from the scratch seemed an unfavorable solution, taking into account limited resources of research projects. In order to find middleware functions, which were supporting our requirements and which could be integrated into our prototype by providing an appropriate application programming interface (API), we conducted an analysis of some promising solutions and evaluated their applicability for our approach.

Hereby, supporting our middleware requirements does not mean, that the respective technology implements the complete desired functionality, but that it supports the realization of it on top of its API. For instance, the support of the "data structure and representation" requirement means, that it is possible to define objects within a technology, e.g., representing sensor data, but not that for all thinkable sensors corresponding objects are already defined.

Thus, the examined technologies should provide mechanisms to realize all the required functions, but not the implementation of the respective functions itself. As potential open accessible technologies for providing at least parts of the required middleware functions, we identified four candidates: OPC-UA, Modbus/TCP, SIP (Session Initiation Protocol) and SOAP with WS-Security.

Table II matches these candidate technologies with the identified functional requirements for the dependability middleware. As result of the comparison of potential technologies we decided for the use of OPC-UA as generic communication and management protocol [51], which seems to provide a good basis to create a generic control architecture.

B. Entities and Roles

In order to realize the intended dependability means, we had to define the respective business logic. As mentioned, these functions may use an underlying infrastructure fulfilling all lower layer requirements and an OPC-UA stack with API as a basis for the new implementation.

As our approach was to provide a complete definition of the conceptual part (yet only implementing selected functions for validation purposes) we had to perform a comprehensive modelling of the desired functions within our dependability domain. For that purpose we had first to define the entities and roles within the DHF. The entities can be identified with the devices participating in the DHF:

- Sensors
- Actuators
- PLCs, DDCs
- PCs
- Mobile devices (smartphones, tablets)
- Active network devices (routers, switches)
- Data storages
- Communication hardware (cables, antennas)
- Embedded systems (Plug PCs, boards)

Sensors and actuators are data sources and sinks respectively; PLCs and DDCs are used for control tasks at automation level, PCs and embedded PCs also for visualization (SCADA), smartphones and tablets the same with less complexity; network devices and communication hardware provide the infrastructure functionality. The entities realize several distinguishable roles, which incorporate the logically independent parts of the whole functionality:

- Client
- Server
- Registrar
- Mediator

The clients (e.g., sensors, PLCs, smartphones) communicate and exchange information with the server. The server (e.g., a PC or embedded board) stores information about the clients and serves thus as a data base. Servers support the possibility to present the information in OPC-UA style. To be allowed to participate in the DHF, all defined parts (clients, servers) must register at the registrar. The registrar provides interfaces for authentication and authorization to the DHF. To communicate with a non-DHF entity, mediators (basically these are gateways, which are able to represent the data structures of the non-DHF part in a DHF compatible manner) map

all relevant information between DHF entities and non-DHF entities.

Network devices (switches, routers, etc.) do not have a functional role regarding the DHF's middleware and are thus considered transparent. To integrate QoS, service classes are defined for the different requirements of the supported applications and triggered by the end systems (clients).

By having defined the roles, the required functionality of the DHF middleware could now be assigned to these roles. In the following subsection, we concentrate on the registrar, as this is the core element for a generic framework, allowing for the integration of multiple clients, servers, and mediators into one framework.

C. Access Rights Management

The main purpose of the X-Model is to enable multiple applications, which are triggered by multiple users, to get access to all DHF devices. This implies the necessity for an access rights management, which is able to assign respective access rights to applications and users, and to enforce the keeping of these access rights. The basic idea is, that the DHF registrar manages the mapping of registered applications and registered devices [36] [32]. Thereto the registrar has not only to provide means to register for new devices and applications respectively to update the registered information for existing ones, but also to decide for appropriate access rights, i.e., it serves as a "Policy Decision Point" (PDP). Figure 7 shows a scheme of the registration process for client and server devices at the registrar.

Of course the access rights assignment can not be performed fully automated, yet a definition of application types respectively user types makes it possible to map access rights not only to individuals, but to groups with similar roles within the system. For instance, flat owners in a house with multiple parties may have less control rights than fire fighters in case of emergencies. These groups need to be assigned the respective access rights only once then. The classification of devices and applications respectively users has still be done manually, thus the system needs a human operator to control the admission to the DHF and to assign appropriate access rights, i.e., the authentication has to be done on a non-technical basis.

Once the registration process is finished, the registered entities are provided with appropriate keys to communicate directly with the peering entity. As every communication has to be encrypted anyway in order to ensure privacy and security of exchanged data, the distribution of decryption keys according to the defined access rights is a way to ensure, that only entities with respective access rights can read this data. This can go

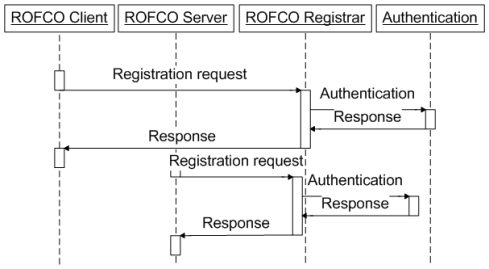

Figure 7. Registration Process

so far, that different entities with different keys can be provided with the same encrypted sensor data, and can decrypt these data with their respective keys in different resolutions, i.e., the different keys represent different authorization to access data. A further authentication with technical means does not have to be performed, as the ownership of the keys is bound to the registration process and therefore secured, provided that the keys are not illegally distributed by the owners.

The big advantage of this solution is the performance, as communication between entities with respective access rights does not require the invocation of central authorities. As resources are especially limited for HA solutions, the concern of performance issues might be unignorable in practice. Yet this is bought by a considerable disadvantage: Key revocation is not possible within such a scheme. The only way to deal with that is to provide access rights only for a defined time, with the necessity to renew admission and thus the key distribution after expiration of granted access rights.

For simplicity reasons, the key distribution may be performed by the registrar [36] [32], which is then also constituting the "Policy Enforcement Point" (PEP). Conceptually, these two functions may also be separated, which could be necessary if performance is still critical, depending on the scale of a DHF realization and the computational power of the registrar device. In opposition to our original intension we decided for symmetric encrpytion for our DHF concept, again for performance issues; only for key exchange asymmetric encryption, i.e., public key encryption, is used.

To complete our security portfolio, further functions have to be addressed: To detect misbehaviour or outages of end systems, keep alive messages are sent during normal operation. Anomaly detection is used to find faulty messages and traffic in the system [49]. With traffic monitoring this traffic can be detected and isolated from the system. Last but not least the triggering of alarms and notifications is not only possible with limit violations from sensor data, but also with peculiar communication attempts.

D. Quality Assurance

To identify potential failures in the design and the application life cycle in the whole DHF and to evaluate potential effects and countermeasures, a Failure Modes and Effects Analysis (FMEA) [52] has been conducted. The FMEA gives an overview about which parts of the DHF are most likely to fail, but also which parts have the most important impacts in case of failures. The FMEA basically consists of following steps:

- A system analysis identifies the parts of the DHF; this can be derived from the design phase, but it also has to take into account external influence factors from the respective system context.
- A function analysis identifies the functions, their allocation to the system parts, and their interoperations and dependencies; thus, critical components can be identified.
- A failure analysis identifies potential failures within the system and allocates them to respective system elements; furthermore, possible reasons for these failures are named.
- A risk analysis identifies the probability of failures, the probability of detection of these failures and the potential impacts; with these factors a risk priority number is calculated and so the most "dangerous" failures can be identified.
- Finally countermeasures and system optimization measures are derived, which shall either minimize the occurrence of failures or the probability of not detecting the failures in time or the potential damage, which could be caused by the considered failure.

Unsurprisingly, outages of controllers have been identified as most dangerous failures, as controllers of HA appliances are not realized in a redundant manner in most cases and thus constituting "single points of failures". The same applies to SCADA systems, yet outages of SCADA systems do not have such immediate consequences, as the several control applications for HA appliances may work autonomously for a certain time. The exact results of the FMEA are documented in the deliverables of the ROFCO project, but not publicly available.

VII. Implementation

As a proof of our concept, we implemented some basic functions prototypically, using the free OPC-UA stack from the OPC foundation [53]. Based on the interests of our partners in the project ROFCO those parts of the framework were implemented, which promised the most direct benefit to them, while still serving as a good basis for our validation process:

1) OPC-UA connection between SCADA system and different control devices (lighting, blinds)
2) Gateway between dependability domain and legacy components
3) Implementation of security (Authentication and Authorization) based on OPC-UA

A. OPC-UA Connection

In our testbed installation, appliances like blinds and lighting were controlled via cTrixs controllers. In order to facilitate communication between SCADA systems and the controllers on OPC-UA basis, parts of the OPC-UA stack had to be implemented at both sides, i.e., the CBUs and the supervising SCADA system. As SCADA system we used Copa-Data's Zenon [48] and the CAPS from cTrixs. Hereby the main focus was to exchange information over an OPC-UA interface by using the OPC-UA information model. Thereto we implemented some selected OPC-UA object types (base object, server objects and the event types). A further focus had been given on the integration of Java and C based OPC-UA libraries into the considered SCADA systems.

B. Legacy Gateway

To interconnect legacy components with the system, it was necessary to map and translate data from the legacy components. Status information about the legacy component had to be stored in an object on the gateway, which represented the properties of the legacy component in the system. The required registration at the system and the mapping of the information exchange had to be handled by the representing object. The implementation of the gateway functionality was based on the OPC-UA ANSI C library and the cTrixs controller communication protocol, which is again based on UDP. As the cTrixs controller was able to map and translate OPC-UA information to EIB/KNX components [54], we used the CBU as our gateway, which controlled the respective EIB appliances (blinds and lighting).

C. Authentication and Authorization

The authentication service was based on an X.509 architecture [55]. The distribution of the key pairs had to be secured by using public key methods to avoid potential leaks in the security concept. The registration and authorization service was supported by an openLDAP infrastructure, which provided a service to register and configure the roles of the different participating devices. The registration of the role and security properties of all devices was stored in an XML configuration file.

Again, the communication between registered users, sensors or gateways is based on the OPC-UA protocol. For legacy devices, encrypted messages can be sent to a gateway by using the OPC-UA communication protocols and interfaces. The gateway can then make a lookup in internal lists or at the registrar in order to decide whether or not to accept the communication from the device. Thus, only messages are accepted, which can be identified by authorized devices.

VIII. VALIDATION

To develop a dependable system, it is a basic precondition to use well established and standardized methods for verification and validation. These methods are based on several different standards, e.g., IEC 61508 [45]. In this paper we concentrate on the validation steps of the ROFCO project. The validation strategy is based on pre-defined use cases, derived from the HA/BA applications Lighting Control and Blinds Control (see Section III). During the course of the project these use cases were adapted to needs and requirements. Thus we have achieved an iterative product life cycle process during the project lifetime in order to enhance the quality of the DHF. The requirement engineering process and the product life cycle process are based on the ISO/IEC 12207-2008 standard [56].

A. Validation Process

As mentioned in Section III, user stories have been used to describe the use case in such a way, that all stakeholders could understand the requirements and the interaction with the DHF. For requirement gathering the verbal description of the use case and the discussion with the stakeholders improved the understanding for the developers. Like in an agile software development process, each single use case had to be validated. Based on the verbal description and the UML Use Case Diagram of each use case we defined the respective test cases. Each test case definition contains attributes, such as verbal test description, pre-conditions, post-conditions, and planned test results, as defined in [57]. The actual test results have been documented in the ROFCO deliverables.

For the exemplary test case Light Maintenance, which is derived directly from the respective use case Light Maintenance as described in Section III, the test case definition looks as follows:

- Test case description: This test case validates the use case Light Maintenance. Thereto the respective application Light Maintenance has to be invoked within the DHF, the status of the lighting in the configured area has to be received, different values (on/off or dimming values) have to be set for single lights and defined groups of lights.
- Pre-conditions: The whole DHF system is installed, the lighting system is installed and configured, the Light Maintenance application is running
- Post-conditions: The Light Maintenance application is still running within the DHF (such that it is possible to re-start this test case several times)
- Planned test results: Status of the lighting is shown correctly, the on/off switches and the dimming controls work correctly for single lights and groups of lights

The test cases form the building blocks of the whole validation process. They have been used in different phases of the validation: During the pre-tests, they helped to identify and fix some misconfigurations in the controller setup and the network configuration. During the final validation trial of the prototype at the Techno-Z Salzburg (see Section V), they have been used to validate

- the control functionality of examined appliances,
- the interworking of the different proprietary HA/BA subsystems, and
- the robustness of the infrastructure and services.

Timing constrains and time criticalities have not been explored so far, yet for future research activities it will be important to address these topics in order to ensure the practical use of the DHF.

B. Validation Results

For both parts of the validation process (pre-tests at module test and integration test level and validation trial at system test level) standardized sets of test cases (“test suites”) have been defined and executed. The test suites have been defined for different parts of the system development process, and are thus constituting an accompanying test process:

- Validation of the developed software components
- Validation of the installed network components
- Validation of the network communication protocols

For instance, the following test suites have been defined for the validation of the network communication protocols:

- Tests of static and dynamic address configurations
- Tests of different routing configurations
- Network link availability tests
- Network device availability tests
- End device reachability tests

Single test cases can be part of one or more of these test suites. For instance, the exemplary test case Light Maintenance is part of the end device reachability test suite. For each test suite, all listed test cases have been executed at least once. If the actual test results were consistent with the planned test results, the test verdict was set to pass, otherwise fail. The test verdict for the whole test suite was set to pass, if and only if all test cases of the test suite achieved positive test verdicts.

Whereas during the pre-test some of the test suites failed, i.e., the respective functionality had to be fixed, the final validation trial yielded only positive test verdicts. Thus, the validation process showed the feasibility of our approach as expected. The interworking of heterogeneous building automation systems based on our X-Model is therefore a potential solution of the mentioned interoperability problems; yet further validation steps are still to be done: First of all, test cases concerning performance and timing issues should be identified and conducted in order to validate the real time capability of the DHF. Furthermore, the definition and execution of test cases derived from the HA/BA application Evacuation Support would help to validate the dependability of the system under test.

IX. Conclusion and Further Work

As a result of our validation trial, we proved the feasibility of our approach, as we were able to access the control devices using different OPC-UA clients. We were able to implement getter and setter functions for the data points of lighting and blinds control in different building units. Furthermore, we developed a dependability concept based on availability calculations according to IEC 61508 [45] functional safety standard and assessed the system relevant risks with an FMEA (see Section VI).

A possible barrier for a wide adoption of our approach in future commercial solutions, especially for the smaller scaled HA market, are the relatively high requirements on the used devices. In order to be able to proceed all the session and rights management data as well as the OPC-UA stack the devices need a certain minimum of computational power; for practical reasons this can not be guaranteed in all cases. Here this can be counteracted by the use of gateways to those legacy systems, which are not able to implement a native OPC-UA connection, yet this limits the beneficiaries of our system to a more narrow system border. However, future developments have to be observed accurately, as the progress of computational power in embedded devices may make this drawback obsolete in a few years. Especially, the market spread of smart phones, which may serve as control devices and user interfaces, brings new chances to HA solutions on IP basis. Another open issue is the influence of the building of communities of households, which will need further research (see Section X). For instance, community based energy optimization applications, but also regulatory aspects, e.g., the EU directive to install smart meters in households (Directive on internal markets 2009/72/EC [58]), may have market implications regarding the use of comprehensive HA systems.

As indicated in [1], the integration of HA/BA appliances with Smart Grids is the main topic of our future research activities. We have just started to test the integration of our X-Model in Smart Grid environments by using Smart Grid applications (like demand response, energy monitoring and health monitoring) with our system approach; yet the challenge will be to ensure the interoperability and collaboration of several HA/BA systems in bigger communities in order to ensure optimization at different scales. Thereto more efforts will be necessary to provide unique control architectures and generic interfaces; moreover the algorithmic side of system optimization (e.g., regarding energy efficiency) has to be addressed in our further research.

Other potential research activities could deal with topics like security and safety. Security will become an even bigger issue than now for two reasons: First, openness requires security means to avoid misuse, and besides all barriers we expect open solutions to spread more widely in future; second, the trend to build communities leads to larger systems with more participants (stakeholders), which exchange privacy and security sensitive data. Safety is already a big issue in BA; if safety solutions get affordable and technically realizable in HA environments, a spread to this market segment is foreseeable, thus research has to deal with this topic.

Industrial solutions can be expected for Mini-SCADA systems on top of dependable frameworks, which do not only provide a one-stop-shop for HA/BA control functionality to the user, but also an easy to use Human Machine Interface (HMI) in order to further increase user-friendliness of HA/BA control. The integration of IP as convergence layer for HA/BA systems is widely accepted in industry now, yet the openness of middleware functions upon IP is still an open issue. Here, the standardization bodies like CEN or ISO are requested to define open standards, which are accepted by the industry; this process is far from being finished.

ACKNOWLEDGMENTS

The work described in this paper was conducted during the project "Robust Facility Communication" (ROFCO), which was funded by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT), and in the "Josef Ressel Zentrum for User-Centric Smart Grid Privacy, Security and Control", which is funded by the Austrian Federal Ministry of Ecomomy, Family and Youth (BMWFJ).

REFERENCES

[1] A. Veichtlbauer, T. Pfeiffenberger, and U. Schrittesser, "Generic control architecture for heterogeneous building automation applications," in *Proceedings of the 6th International Conference on Sensor Technologies and Applications (SensorComm 2012)*, Rome, August 2012, pp. 148–153.

[2] K. Charatsis, A. Kalogeras, M. Georgoudakis, J. Gialelis, and G. Papadopoulos, "Home / Building Automation Environment Architecture Enabling Interoperability, Flexibility and Reusability," in *Proceedings of the IEEE International Symposium on Industrial Electronics 2005 (ISIE 2005)*, vol. 4, Jun. 2005, pp. 1441–1446.

[3] F. Ferreira, A. Osorio, J. Calado, and C. Pedro, "Building Automation Interoperability – A Review," in *Proceedings of the 17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010)*, 2010, pp. 158–161.

[4] M. Ciesielska and F. Li, "The connected home: From market barriers to business model solutions," in *Building the e-World Ecosystem*, ser. IFIP Advances in Information and Communication Technology, T. Skersys, R. Butleris, L. Nemuraite, and R. Suomi, Eds. Springer Verlag, Oct. 2011, vol. 353, pp. 189–199.

[5] *Smart Grid Reference Architecture*, CEN/Cenelec/ETSI Smart Grid Coordination Group Std., Nov. 2012.

[6] *ISO/IEC 7498-1:1994 Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*, International Standards Orginization (ISO) Std., 1994, Accessed: 2013-02-25. [Online]. Available: http://www.iso.org/iso/catalogue_detail. htm?csnumber=20269

[7] J. Postel, *Internet Protocol – DARPA Internet Program Protocol Specification*, RFC 791, IETF Std., Sep. 1981.

[8] Salzburg Research Forschungsgesellschaft. (2012) ROFCO – Robust Facility Communication. Accessed: 2012-06-19. [Online]. Available: http://www.salzburgresearch.at/en/projekt/rofco_en/

[9] Salzburg University of Applied Sciences. (2013) en-trust – Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control. Accessed: 2013-02-04. [Online]. Available: http://www.en-trust.at/

[10] A. Veichtlbauer, D. Engel, F. Knirsch, O. Langthaler, and F. Moser, "Advanced metering and data access infrastructures in smart grid environments," in *Proceedings of the 7th International Conference on Sensor Technologies and Applications (SensorComm 2013)*, Barcelona, Aug. 2013, (accepted).

[11] M. S. Jimenez, *Smart Grid Mandate*, European Commission Directorate-General for Energy Std., 2012.

[12] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. Vol. 1 No.1, pp. 11–33, 2004.

[13] J. P. Thomesse, "Fieldbuses and interoperability," *Control Engineering Practice*, vol. 7, iss. 1, pp. 81–94, Jan. 1999.

[14] E. Finch, "Is IP everywhere the way ahead for building automation?" *Facilities*, vol. 19, iss. 11/12, pp. 396–403, 2001.

[15] Siemens AG. (2011) Total Building Solutions für intelligente Gebäude – Siemens Building Technologies. Accessed: 2012-06-19. [Online]. Available: http://www.industry.siemens.de/buildingtechnologies/de/de/total-building-solutions/Seiten/total-building-solutions.aspx

[16] ABB Asea Brown Boveri Ltd. (2012) Raumtalk – Building Automation over IP. Accessed: 2012-04-11. [Online]. Available: http://www.abb.at/cawp/deabb201/24d156e58bc98443c125720b0025238d.aspx

[17] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus, "A Modular Architecture for Building Automation Systems," in *Proceedings of the 6th IEEE International Workshop on Factory Communication Systems (WFCS '06)*, Jun. 2006, pp. 99–102.

[18] American Society of Heating, Refrigerating and Air-Conditioning Engineers Inc., "BACnet - A Data Communication Protocol for Building Automation and Control Networks," ANSI/ASHRAE Standard 135-2004, 2004.

[19] D. Snoonian, "Smart buildings," *Spectrum, IEEE*, vol. 40, pp. 18–23, Aug. 2003.

[20] U. Schrittesser, "Synthese von redundanten vermaschten wlan," Master's thesis, Salzburg University of Applied Sciences, Jun. 2008, in German.

[21] CAS. (2010) OPC Unified Architecture. Accessed: 2012-06-19. [Online]. Available: http://www.commsvr.com/UAModelDesigner/Index.aspx

[22] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*. Springer-Verlag Berlin Heidelberg, 2009.

[23] J. Postel, *Transmission Control Protocol – DARPA Internet Program Protocol Specification*, RFC 793, IETF Std., 1981.

[24] A. Fernbach, W. Granzer, and W. Kastner, "Interoperability at the Management Level of Building Automation Systems: A Case Study for BACnet and OPC UA," in *Proceedings of the 16th IEEE Conference on Emerging Technologies and Factory Automation (ETFA '11*, Sep. 2011.

[25] RESI Informatik & Automation GmbH. (2013) Resi SCADA 2D. Accessed: 2013-02-25. [Online]. Available: http://www.resi.cc/wordpress/prestashop/product.php?id_product=59

[26] ETM Professional Control GmbH. (2013) Simatic WinCC Open Architecture. Accessed: 2013-02-25. [Online]. Available: http://www.etm.at/index_e.asp?id=2&sb1=&sb2=&sb3=&sname=&sid=&seite_id=6

[27] Cooper Power Systems. (2010, Oct.) Mini-SCADA solution. Accessed: 2013-02-04. [Online]. Available: http://www.cooperindustries.com/content/dam/public/powersystems/resources/library/1100_EAS/B110007341.pdf

[28] P. E. Rovsing, P. G. Larsen, T. S. Toftegaard, and D. Lux, "A reality check on home automation technologies," *Journal of Green Engineering*, pp. 303–327, 2011.

[29] M. Tariq, Z. Zhou, J. Wu, M. Macuha, and T. Sato, "Smart grid standards for home and building automation," in *Proceedings of the 2012 IEEE International Conference on Power System Technology (POWERCON 2012)*, 2012.

[30] T. S. Hjorth and R. Torbensen, "Trusted domain: A security platform for home automation," *Computers & Security*, vol. 31, no. Issue 8, pp. 940–955, Nov. 2012.

[31] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus, "Security in Networked Building Automation Systems," in *Proceedings of the 6th IEEE International Workshop on Factory Communication Systems (WFCS '06)*, Torino, Jun. 2006, pp. 283–292.

[32] C. Probst and A. Veichtlbauer, "Security Features of a Generic Sensor Data Acquisition System," in *Proceedings of the 6th International Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet 2010)*, Bodrum, Turkey, Oct. 2010, pp. 78–81.

[33] T. I. Salsbury, "A Survey of Control Technologies in the Building Automation Industry," in *Proceedings of the 16th IFAC World Congress*, vol. 16, part 1, Prague, Czech Republic, Jul. 2005.

[34] S. Makarechi and R. Kangari, "Research methodology for building automation performance index," *International Journal of Facility Management*, vol. 2, no. 1, 2011.

[35] A. Veichtlbauer and T. Pfeiffenberger, "Dynamic evacuation guidance as safety critical application in building automation," in *Proceedings of the 6th International Conference on Critical Information Infrastructure Security (Critis 2011)*, Lucerne, Switzerland, Sep. 2011.

[36] C. Probst, "Konzeptionierung eines Benutzermanagements für den Zugriff auf vertrauliche Daten von IP fähigen Sensornetzen," Master's thesis, University of Applied Sciences Salzburg, May 2010, in German.

[37] International Telecommunication Union. (2008) X.501. Accessed: 2012-06-19. [Online]. Available: http://www.itu.int/rec/T-REC-X.501

[38] C. Rigney, A. C. Rubens, W. A. Simpson, and S. Willens, *Remote Authentication Dial In User Service (RADIUS)*, RFC 2865, IETF Std., Jun. 2000.

[39] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, *Diameter Base Protocol*, RFC 3588, IETF Std., Sep. 2003.

[40] Cisco Systems, Inc. (2013, Feb.) Cisco Systems, Inc. Cisco Systems, Inc. Accessed: 2013-02-25. [Online]. Available: http://www.cisco.com/

[41] R. Baumann, S. Heimlicher, M. Strasser, and A. Weibel, "A survey on routing metrics," Computer Engineering and Networks Laboratory, ETH-Zentrum, Switzerland, Tech. Rep., Feb. 2007, accessed: 2012-06-19. [Online]. Available: http://www.baumann.info/public/tik262.pdf

[42] S. Shao, M. Pipattanasomporn, and S. Rahman, "Grid integration of electric vehicles and demand response with customer choice," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 543–550, Mar. 2012.

[43] U. Greveler, B. Justus, and D. Löhr, "Multimedia content identification through smart meter power usage profiles," in *Proceedings of the 2012 International Conference on Information and Knowledge Engineering (IKE'12)*, Las Vegas, USA, Jul. 2012, pp. 383–390.

[44] G. Panholzer, A. Veichtlbauer, P. Dorfinger, and U. Schrittesser, "Simulation of a robust communication protocol for sensor data acquisition," in *Proceedings of the 6th International Conference on Wireless and Mobile Communications (ICWMC 2010)*, Valencia, Spain, Sep. 2010, pp. 145–150.

[45] *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems*, International Electrotechnical Commission (IEC) Std., Apr. 2010.

[46] K. Werthschulte, "Integration von heterogenen Bussystemen in die Heimautomatisierung unter Verwendung von Middleware," Ph.D. dissertation, Technical University Munich, 2003.

[47] C. Busemann, C. Kuka, U. Westermann, S. Boll, and D. Nicklas, "Scampi – sensor configuration and aggregation middleware for multi platform interchange," in *Proceedings of the 39th Annual Conference of the Society for Informatics*, Lübeck, 2009.

[48] Ing. Punzenberger COPA-DATA GmbH. (2013) HMI SCADA Software zenon by COPA-DATA. Accessed: 2013-02-25. [Online]. Available: http://www.copadata.com/en/home.html

[49] Quanmax AG. (2013) MF Security Gateway. Accessed: 2013-02-25. [Online]. Available: http://www.underground8.com/de/products/mf_security_gateway.html

[50] cTrixs GmbH. (2012) CBU cTrixs Base Unit. Accessed: 2013-03-14. [Online]. Available: http://www.ctrixs.com/systemubersicht

[51] M. Melik-Merkumians1, T. Baier, M. Steinegger, W. Lepuschitz, I. Hegny, and A. Zoitl, "Towards OPC UA as portable SOA Middleware between Control Software and External Added Value Applications," in *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies and Factory Automation (ETFA 2012)*, Krakow, Sep. 2012.

[52] T. Tietjen, D. Müller, and A. Decker, *FMEA Praxis – Das Komplettpaket für Training und Anwendung*, 3rd ed. Carl Hanser Verlag, Mar. 2011, in German.

[53] OPC Foundation. (2012) OPC – The Interoperability Standard for Industrial Automation & Other. Accessed: 2012-06-19. [Online]. Available: http://www.opcfoundation.org

[54] *ISO/IEC 14543-3:2006 Information technology – Home Electronic Systems (HES) Architecture*, International Standards Orginization (ISO) Std., 2006, Accessed: 2013-03-15. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=43364

[55] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 5280, IETF Std., May 2008.

[56] *ISO/IEC 12207-2008 - ISO/IEC/IEEE Standard for Systems and Software Engineering - Software Life Cycle Processes*, IEEE Std., Jan. 2008, Accessed: 2013-02-21. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=4475822

[57] *IEEE 829-2008 - IEEE Standard for Software and System Test Documentation*, IEEE Std., Jul. 2008, Accessed: 2013-02-21. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=4578271

[58] The European Parliament and the Council of the European Union, "Directive 2009/72/ec," *Official Journal of the European Union*, vol. L 211, pp. 55–93, Aug. 2009. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:211:0055:0093:EN:PDF