BIM4EEB

BIM based fast toolkit for Efficient rEnovation in Buildings

D6.1 Open format and formalised requirements specification for procurement

This project has received funding from European Union’s H2020 research and innovation programme under grant agreement N. 820660

Program	H2020
Call for Proposal	LC-EEB-02-2018 Building information modelling adapted to efficient renovation
Project Title	BIM based fast toolkit for Efficient rEnovation in Buildings
Acronym	BIM4EEB
Project Grant Agreement	820660

D6.1 Open format and formalised requirements specification for procurement

Work Package	WP6
Lead Partner	University College Cork (UCC)
Contributing Partner(s)	Technical University of Dresden (TUD)
Dissemination Level	Public
Type	OTHER: System specifications
Due date	31/12/2019
First Submission Date	20/12/2019
Updated version submission Date	31/05/2021
Version	2.0 FINAL

DOCUMENT HISTORY

Version	Date	Comments	Main Authors
0.1	23.07.2019	ToC created for review	Dr. Andriy Hryshchenko (UCC)
0.2	09.09.2019	Add first outline/plan for chapter 3 & 4	Matthias Lehmann (TUD)
0.3	22.10.2019	Added content in ch. 3 & 4	Matthias Lehmann (TUD)
0.4	05.11.2019	Added content in 2.1	Mai Tuan Linh (TUD)
0.5	06.11.2019	Added content in ch. 1 & 2 restructuring of the document	Matthias Lehmann (TUD)
0.6	08.11.2019	Added missing contents, cleaning the document	Matthias Lehmann (TUD)
0.7	10.12.2019	Formatting and corrections	Dr. Andriy Hryshchenko (UCC)
0.8	12.12.2019	Additional corrections	Dr. Andriy Hryshchenko (UCC)
0.9	16.12.2019	Proofread, adjustments, corrections	Dr. Dominic O'Sullivan (UCC)
0.10	19.12.2019	Adjustments	Matthias Lehmann (TUD)
0.11	20.12.2019	Final adjustments and corrections. Submission to Project coordinator for approval.	Dr. Andriy Hryshchenko (UCC)
1.0 FINAL	20.12.2019	Submitted to the BIM4EEB Project reviewer	Project Coordinator
1.01	10.02.2021	Initial changes w.r.t. remediation plan	Dr. Andriy Hryshchenko (UCC)
1.02	12.04.2021	Corrections and improvements of references as per comments received	Dr. Andriy Hryshchenko (UCC)
1.5 FINAL	19.04.2021	Final polishing of V2.0 report	Dr. Andriy Hryshchenko (UCC), Brian O'Regan (IERC, UCC)
1.6	05.05.2021	Correction and updates. Review of PoliMi Team	Bruno Daniotti (Polimi)
1.7	27.05.2021	Plagiarism check and final polishing	Dr. Andriy Hryshchenko (UCC)
2.0 FINAL	31.05.2021	Final updates and submission	Bruno Daniotti (Polimi)

Statement of originality:
This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. BIM4EEB action has received funding from the European Union under grant agreement number 820660.

EXECUTIVE SUMMARY

The planning process, which is needed for procurement of a system, is always a very error-prone process due largely to the fact that there are many stakeholders are involved. These stakeholders with their individual business relationships and activities, in terms defined for the BIM4EEB project, have been defined in detail within the work done in other work-packages (see D2.1 Tables 4-25, Figure 5; D3.3 Tables 1, 5-8).

Moreover, in the detailing and planning of a building renovation, taking for example Building Automation Systems, there are many different stakeholders like electricians, system integrators, heating engineers and a lot of more involved. The large diversity of trades in this sector challenges planners to deliver combined systems to ensure buildings can be more energy efficienct and comfortable after renovation. An example of the various types of interactions that can occur would be that a heater should shut off in case where an occupant of a building opens a window.

Dedicated to the field of Room Automation Systems, as part of Building Automation Systems, a National guideline (VDI3813:2011-05, 2011) is available to describe required functionality semantically. This guideline is currently worldwide the only basis for the semantical description of functionality in this sector.

The Technical University of Dresden (TUD) has formerly researched and implemented a semi-automated design process for Room Automation Systems, based on this VDI3813, which resulted in the tool suite AUTERAS (Technische Universität Dresden, 2013). This tool suite is enhanced in the BIM4EEB WP6 to accomplish the goals and terms of our research project.

Firstly, the basis of describing functionalities semantically is introduced in this deliverable as well as how to define functional requirements to Room Automation Systems.

After this, the non-functional requirements to Room Automation Systems and their components are described.

The last part of this report refers to the decription of a system model of the Room Automation Systems, which can be used as an input by the following parts of the BIM4EEB's tools.

PUBLISHING SUMMARY

Before the procurement of components of a system, the system must be planned as technology- and manufacturer- independent.

The planning process of new building, as well as their renovation, is a very complex procedure. As mentioned, many different trades have to work together, while different terms and notations impede these collaborations. This has an impact not only on complete processes' planning, but also on the resultant effectiveness of the Building Automation Systems. Electricians, system integrators, lighting experts, heating engineers and many more professionals have to work very closely together in order to achieve a result in the form of energy-efficient buildings with a high level of comfort.

The tool suite AUTERAS (Technische Universität Dresden, 2013) supports planners to design Room Automation Systems (as part of Building Automation Systems) with a semi-automated process of functional requirement survey and generation of functional block-based designs, which use standardized symbols to ensure a high comprehension from stakeholders in different trades. The resulting designs can be used directly to form bills of quantities for the procurement process.

Therefore, the specification of functional and non-functional requirements is explained in this D6.1 report with use of current AUTERAS-based high-end knowledge and achievements. To ensure a high energy efficiency of the resulting Room Automation Systems, special attention was taken to the consideration of Energy Efficiency Classes as an example for non-functional requirements.

Furthermore, the common web-based system model for the designs of the Room Automation Systems was established, so that other subsequent stakeholders will be able to re-use them directly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Background

The procurement process of buildings' constructing / renovating is a very complex and error-prone process. As previously demonstrated in the BIM4EEB WPs 2 and 3 reports, there are many steps before a tendering process can be started with many stakeholders are involved until then. When using engineering services' terminology to describe engineering services for building, infrastructure and industrial facilities (as per (BS EN 16310, 2013)) based on D2.1 and D3.3 reports, the Stages of Buildings' Lifecycle (BLS) until the procurement are defined as the following:

0. Initiative;
1. Initiation;
2. Design;
3. Procurement.

In the German HOAI (HOAI:2013-07, 2013) – "Honorarordnung für Architekten und Ingenieure" (English: Federal Honorarium Code for Architects and Engineers) – these steps are more compartmentalized.

In the following table these steps as well as the minimum results will be shown.

Table 1: Project phases and tasks for project leader (from HOAI)

phase (EN16310)	phase (HOAI)	minimum results (HOAI)
initiative	-	-
initiation	base estimates	• clarification of the task on the basis of the specifications or the requirements planning of the client • local inspection • advise on the entire performance and investigation needs • formulate the decision-making aids for the selection of other people involved in planning • combine, explain and document the results
	preliminary planning (project and planning preparation)	• analysing the basics, matching the services with those involved in the planning process • coordinating the objectives, pointing to conflicting goals • preparing the preliminary planning; examining, presenting and evaluating variants according to the same requirements; drawings on a scale according to the type and size of the object • clarification and explanation of the essential relations, specifications and conditions (for example, urban planning, design, functional, technical, economic, ecological, building physics, energy management, social, public law) • providing the work results as a basis for the other parties involved in the planning process as well as the coordination and integration of their services • preliminary negotiations on the approvability • cost estimate according to DIN 276, comparison with the financial boundary conditions • creating a schedule with the essentials of the planning and construction process • summarizing, explaining and documenting the results
design	design planning (system and integration planning)	• elaboration of the design planning, taking into account the essential relations, specifications and conditions (for example urban planning, design, functional, technical, economic, ecological, social, public law) on the basis of preliminary planning and as a basis for the further phases of the work and the necessary public-law approvals using the contributions of others involved in the planning. drawings according to the type and size of the object to the required comprehensiveness and level of detail taking into account all technical requirements, for example for buildings on a scale of 1:100, for example, indoors on a scale of 1:50 to 1:20 • providing the work results as a basis for the other parties involved in the planning as well as the coordination and integration of their services • object description • negotiations on the approvability • cost calculation according to DIN 276 and comparison with the cost estimate • updating the schedule • summarizing, explaining and documenting the results

	approval planning	• preparation and compilation of applications and evidence for public authorizations or approvals, including requests for exemptions and liberations, as well as necessary negotiations with authorities using the contributions of other parties involved in planning • submitting the applications • complete and adapt the planning documents, descriptions and calculations
	detailed design	• developing the execution planning with all the details necessary for the execution (drawing and text) on the basis of the design and approval planning up to the solution ready for execution, as a basis for the further work phases • execution, detail and construction drawings by type and size of the object to the required extent and degree of detail, taking into account all technical requirements, for example in buildings in scale 1:50 to 1:1, for example in interiors in scale 1:20 to 1:1 • providing the work results as a basis for the other parties involved in the planning, as well as the coordination and integration of their services • updating the schedule • updating the detailed planning because of the trade-oriented processing during object execution • checking necessary assembly plans of the building plans planned by the object planner and constructional components for conformity with the execution planning
procurement	preparation of the awarding of contracts	• setting up an appointment schedule • the drawing up of specifications with bills by service sectors, determination and aggregation of batches on the basis of the design planning, using the contributions of others involved in the planning of technical experts • reconcile and coordinate the interfaces to the service descriptions of those involved in the planning process • determine costs on the basis of planner-based bills of quantities with costs • cost control by comparing the bills priced by the planner with the cost calculation • compilation of the tender documents for all performance areas
	Participation in the award	• coordinating the awards of the specialist planners • obtaining offers • reviewing and evaluating offers, including the establishment of a price breakdown by item or sub-service, checking and evaluating the offers of additional and changed services by the performing companies and the reasonableness of the prizes • conducting bidders' talks • preparation of procurement proposals, documentation of the award procedure • compiling the contract documents for all service areas • compare the tendering results with the bills of quantities priced by the planner or the cost calculation • participation in the award of the contract

The essence of the Table 1 above is that for procurement in building construction and renovation a lot of planning and design tasks must be done. Especially for larger buildings, where many stakeholders/trades are involved, it is necessary that plans of different trades will be merged and harmonized.

In the construction domain for buildings itself, a trade-spanning process is relatively well-advanced and is supported by different BIM-methods and their implementations. A large problem area exists in the domain of building services. This problem has historically evolved, from a period when different services were agnostic, and the systems themselves were small. However, over time these building systems increased in complexity and their interactions and inter-dependencies became more relevant, especially nowadays in the age of the “Internet of Things”.

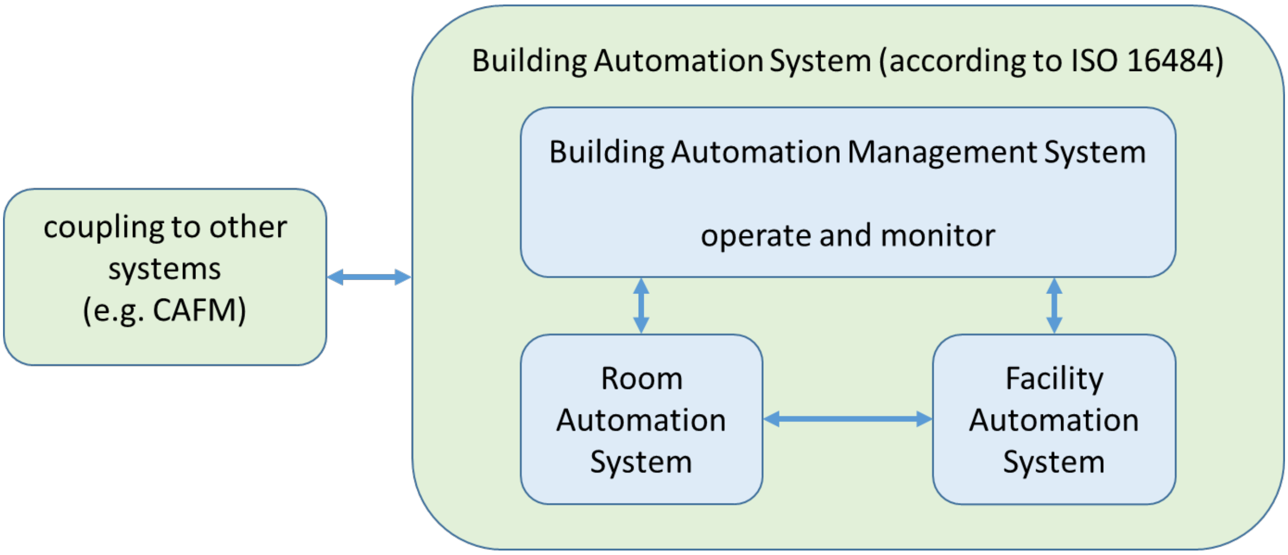

Figure 1 : Relation of the different parts of building automation systems according to ISO 16484

The Figure 1 above shows the relationship between the different parts of a Building Automation Systems (BAS). The basis to describe a BAS was formed in the German VDI3814 guideline (VDI3814:2017-07, 2017). This guideline was later transferred into the standard ISO16484 (ISO 16484-2:2004, 2004). In the current edition the functions (sensors, actuators, pumps, fans etc.) of the parts are described in text form on a very high level of detail. However, the application functions (which are most important part of room automation) were not considered. This can be rectified by establishing the standard mentioned (VDI3813:2011-05, 2011), which defines all the functions in the field of room automation (for HVAC, lighting and shading) with the description method of function blocks (further explained in next chapter). Currently the (VDI3814:2017-07, 2017) is under construction to establish this function block based methodology also in this norm and to re-describe the existing functions and to include the (VDI3813:2011-05, 2011) functions into the same norm. It is planned to transfer the new (VDI3814:2017-07, 2017) into a new version of the ISO16484 so that it will become an international standard.

The larger problem identified is that available device-description methodologies currently embrace only the geometrical and material information (e.g. connectors). Based on German standards, this methodology is described in VDI3805 standard (VDI3805:2011-10, 2011), partwise also in (ISO 16757-1:2015, 2015). The recommended description format there is International Foundation Classes - IFC (based on (ISO 16739-1, 2018), evolved from (International Organization for Standardization, 2013)). This is very useful format for the addressed information (mainly geometry, material and general functionality), but it is not sufficient to describe flows of data or events. This is not possible with IFC and also not planned to include.

The group for Technical Information Systems (TIS) of TUD, as previously mentioned, have developed the planning tool suite for Room Automation Systems (RAS): “AUTERAS – AUTomated Engineering of Room Automation Systems”, (Technische Universität Dresden, 2013). This tool suite allows defining functional user-requirements to the automation system in a building. These functional requirements will be transformed to a function-block-based functional design, which describes the functionality as well as the relations between the different functions in an unambiguous way. Moreover, these functional designs are, platform- and vendor- independent which is one of their main advantages. This allows using these designs in bidding processes. The functional designs can be used now to search devices which can fulfil required functions in the functional design and are interoperable together. With the result of this step a system integrator is able to buy those needed devices and integrate them into the building and related controlling systems. In the following report of WP6 D6.2, the workflow (for an overview see Figure 2 overleaf) will be described more in detail.

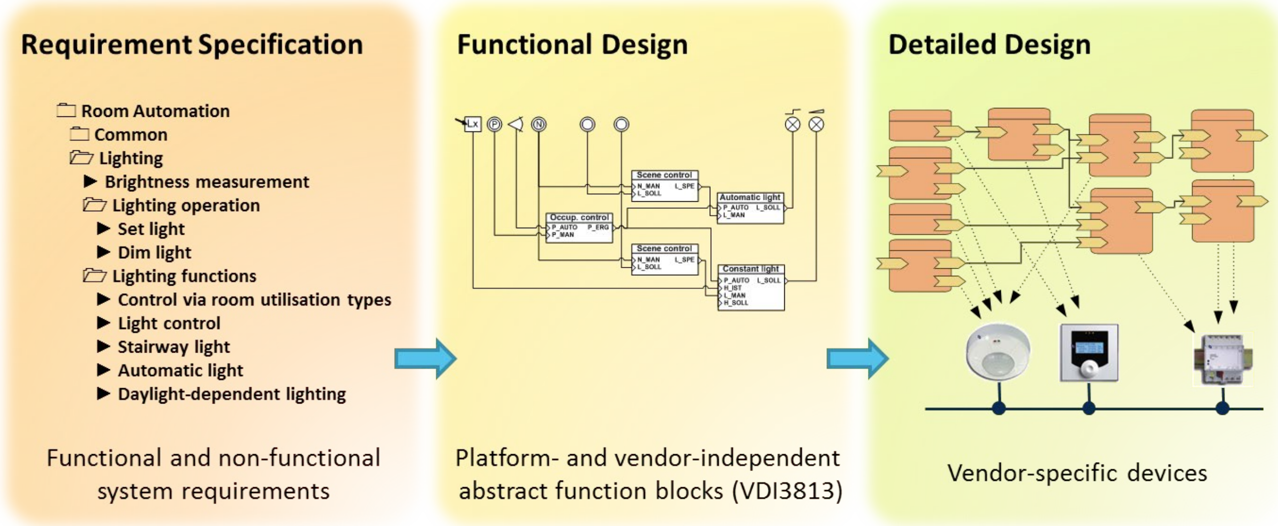

Figure 2: AUTERAS RAS-planning workflow

To provide this functionality, AUTERAS consists of 5 tools:

- **AUTERASplan:** tool to define functional requirements and generation of functional designs
- **AUTERASdesign:** tool to search devices from AUTERAScatalog to fulfil functional designs
- **AUTERAScatalog:** device definition store for semantical device descriptions
- **AUTERASstore:** web-based store for AUTERAS projects
- **AUTERASlite:** lightweight variant of AUTERASplan/design

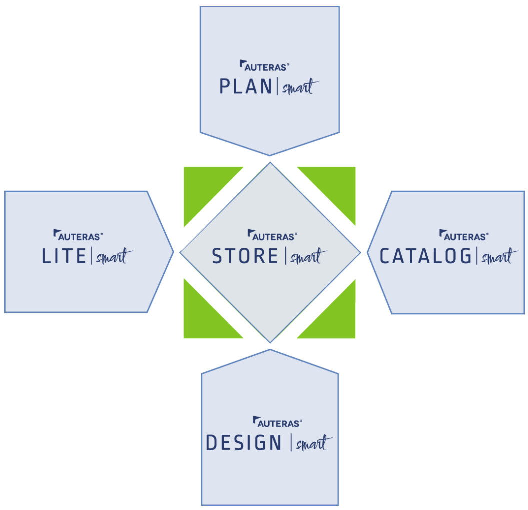

Figure 3: Parts of AUTERAS

This tool suite is able to cover the problem area of functional requirements as well as the usage of them later on. In this deliverable it is described how AUTERAS was enhanced in WP 6 with the field of non-functional requirements to fulfil the objectives of BIM4EEB. AUTERAS was researched and developed in several national and European projects, like (SCUBA, 2011-2014) - (FP7 – GA-no: 288079) and (TOPAs, 2015-2018) - (H2020 – GA-No: 676760).

AUTERAS' is strongly based on the German norm (VDI3813:2011-05, 2011) – the worldwide first semantical description methodology for the functionality of RAS. Therefore, the usage of AUTERAS is recommended in BIM4EEB to describe the functionality installed with RA devices in buildings or to plan new buildings.

1.2 Involvement of Stakeholders in Procurement procedures

Procurement is an internationally recognized profession. Organisations aim at getting the best value product or service by encouraging openness, developing the relationships with potential suppliers, engaging with service users and marketing opportunities widely. The purchasing profession has been gaining recognition in developed countries faster than in the developing countries (Matechak, 2009). Worldwide, public procurement has become an issue of public attention and debate, and has been subjected to reforms, restructuring, rules and regulations. Public procurement refers to the acquisition of goods, services and works by a procuring entity using public funds (World_Bank, 2019). Procurement process starts when an entity has identified a need and decided on its procurement requirement. Procurement goes through the process of risk assessment, seeking and evaluating alternative solutions, contract award, delivery of, and payment for the property and/or services and, where relevant, the ongoing management of a contract and consideration of options related to the contract. Procurement also extends to the ultimate disposal of property at the end of its useful life. (Manu, 2005) – *reference confirmed.*

Furthermore, the procurement function is responsible on one hand for the identification of the end-user's needs and, by utilizing suppliers, meeting them. Therefore, by its very nature, procurement is a "service" function. (Ishola, 2010) In this study, procurement is defined as service functions provided by a dedicated team of professionals operating at the interface between the organizations' suppliers and the end-user department (s) in order to effectively and efficiently meet the supply needs of the organization. According to (Ellaram et al.,1989), in procurement the customer service outcome exists in two domains - the supplier activity domain and the end-user response domain. Thus, the study identifies two customers to the procurement function: internal and external i.e. the end users and the suppliers respectively. In a procurement process therefore, efforts must be dedicated to ensuring the complete satisfaction of not only the end-user or customer of a product and/or service, but also the satisfaction of the suppliers whose products or service are incorporated into the end-user /customer order and whose performance impacts the end user satisfaction (Gordon, 2009). Also available at (Gatta, 2017).

1.2.1 **Five key principles to adhere to in stakeholder engagement**

There are five vital principles for procurement professionals wanting positive stakeholder engagement:

1. Identification of stakeholders: It is critical that in procurement, individuals can identify who the key stakeholders are within the business. By getting to know them, you can get a clear understanding of what they want, care for and influence across the organisation.
2. Active and early engagement: Essentially, the earlier you can engage with the relevant stakeholders the better the outcome. As part of a strategic procurement plan, you naturally gain their participation in negotiations. Senior board members will have a higher level of influence than someone at the junior end. So, by getting the more experienced individual on board at an early stage, you can utilise their ideas quicker and with more efficiency. In the procurement environment, this is particularly true in driving change management and transformation.
3. Listening to stakeholders: We believe it's critical that procurement teams listen to their stakeholders in order to manage their expectations. From a procurement perspective, acknowledging and then being able to act on stakeholder input is key to driving success. If the project is on brief, then there is every chance that the client or customer will be pleased with the result.
4. Communication: We are sure that communication with stakeholders needs to be concise and clear. Having a positive working relationship with stakeholder's means the output will be of higher value and their interest in the project more significant. When there are changes in a project, stakeholders must feel like they're a part of the decision making process and are adding value.
5. Creating value and ensuring success: A procurement function that educates and works with stakeholders is able to create value for both parties, and best practice is a natural way to this type of success.

Abovementioned statements are presented as a shared view of the BIM4EEB WP6 partner's working approaches and based on (Pegg, 2015) open-source publication.

1.2.2 **Procurement and its relation to Renovation processes**

There are several good sources that can be quoted to highlight this topic. The SCI-Network, "Sustainable Construction and Innovation through Procurement" is one of them to follow-up. This European Network of public authorities was established through a project co-funded by the European Commission's CIP programme through the Lead Market Initiative. (SCI-Network, 2009-2012)

Accordingly to this Network' initiatives and reports, the procurement process in a renovation project is the same as that of a new building. For a typical construction project the process starts with a requirement from the public authority for a constructed asset. Design work may be undertaken either in-house or by external consultants appointed following a competitive process. In some cases these external consultants undertake a tendering process to select a main contractor, whereas in other cases this is done by the client authority directly. The public client can also decide to combine both design and the actual construction and procure a design and build- contractor, or even design, build and operate.

In most cases, similarly to the outcomes of the BIM4EEB WP2 reports, the main contractor will take care of employment of subcontractors and the procurement of materials. After successful completion of the project there will be a hand over of the completed asset to the end user. In the case of renovation projects there is also the possibility to require a managed service for instance as an energy service contract. The contractor then designs, conducts the renovation and maintains responsibility for the energy savings and other required quality targets during the operation of the contract period.

Sustainable refurbishment of an existing building requires relevant data on the current environmental performance in order to set the appropriate target level for the refurbishment. The energy certificate of buildings is one basis of information (if available). Audits may need to be carried out, which may require contracting a qualified auditor. The audit usually points out the most cost-effective measures to achieve energy- and resource efficiency.

In some EU member states, for instance in Finland, government funding is available for energy auditing in public sector buildings. If an audit is ordered, it is important for the public authority to set clear objectives and requirements on this as well, so that it serves the purpose of the forthcoming renovation processes.

Table 2 (overleaf) provides a mapping of stakeholders' activities (as defined in the report D2.1, Tables 4-26) to Building Lifecycle Stages (BLS) as summarised in the D3.3 report. It illustrates that numerous stakeholders are not involved in the defined use cases w.r.t procurement activities.

All this information will be used for the following development of this report.

Table 2: Stakeholders and their involvement in Use Cases 1-3 w.r.t. procurement activities

Activities as per D2.1	Building Lifecycle Stages: BLS 0-6						
	0 Initiative	**1 Initiation**	**2 Design**	**3 Procurement**	**4 Construction**	**5 Use**	**6 End of Life**
Use Case 1	3, 4	13, 14, 16, 20, 21, 32, 47, 48	55, 56, 60, 61, 73-75, 95, 96, 118	29, 30	-	-	178, 184, 185,
Use Case 2	-	17, 18, 19, 27	63-68, 76-79, 83, 84, 86-91, 97-105, 107-110, 119-129, 131, 132, 134, 135	28, 30, 31	141, 142, 160, 161	-	181-183, 187, 191, 195
Use Case 3	-	22, 28-30, 33-35, 37, 40, 44	111, 112, 116, 117	137, 138, 164,175, 177, 181, 190	143-148, 155, 156, 159	166-170, 173-175	189, 192

1.3 Scope of this report

The scope of this document is to describe methods and methodologies involved for definition of requirements (functional and non-functional) for a Room Automation System (RAS). The reason for limiting focus to RAS is simply because of the volume of problems that occur in this subset of automation with respect to describing their requirements. For the other parts of Automation Systems there are suitable methodologies and formats available on the market, like (eCl@ss, 2019) for example, for physical description of components.

Therefore, in the next chapter a functional requirement specification will be introduced. This new specification replaces the old static implementation in AUTERAS. It is now possible to integrate easily new function blocks and conditions to requirements to enhance the functional capabilities of AUTERAS. Additionally, there will be described how these functional requirements can be displayed as functional descriptions according to (VDI3813:2011-05, 2011). The following Chapter 3 is dedicated to non-functional requirements to the RAS. Furthermore, the Chapter 4 brings collected information together and describes at first a functional device description format based on XML (W3C, 2021) and after that a web-based system model for RAS projects.

1.4 Relevance to other deliverables

This report D6.1 introduces a new aspect of the BIM4EEB objectives; moreover, it influences several other subsequent deliverables, i.e.:

- D6.2: “Methods and tools for selecting devices and linking them to the generic model”

  D6.2 uses the requirement definition methodologies defined in this deliverable to integrate them in a semi-automated design process for planning Room Automation Systems.

- D6.3: “Tool for constraint checking BAC-topologies vs building codes”

  Methodologies for checking the fulfilment the requirements, defined in this deliverable, will be described in D6.3.

Nevertheless, the data models/stores/interfaces used for the tools and methods described in D6.1 must be compatible to the other tools and data structures of BIM4EEB to enhance existing data respectively to provide data for the other tools (mainly information about the Room Automation Systems) in the buildings. Therefore, this deliverable must strongly consider the following previous architectural deliverables:

- D2.1: “Definition of relevant activities and involved stakeholders in actual and efficient renovation processes”
- D3.1: “A BIM-based framework for building renovation using the linked data approach and ontologies”
- D3.2: “A refined, integrated domain ontology for occupants, building services, building energy and acoustics”

2 Functional Requirements

Building Automation and Controls (BAC) are a combination of hardware and software that control a building's integrated systems. This chapter is dedicated to the description of functional requirements of Building Automation Systems (BAS). Therefore, in the first section of this chapter a semantical description methodology of BAS-functions is described. In the second section the methodology to gather functional requirements and to ensure their validity is explained.

2.1 Semantical description method for functional requirements in BAS

In practice, the functional requirements to a building automation system will be described mostly as a textual description. This approach has some important disadvantages. Mainly the requirement descriptions are very vague and unclear. Different persons can interpret them in very different ways. Another problem is the legal liability in case of mistakes. If the requirement description is vague, the planner is maybe able to argue that omissions not a mistake. Many plans are also made graphically. However, in practice the most planners use for different symbols for the same requirements. Therefore, there is also a great potential for misunderstandings.

These misunderstandings of the current textual and graphical methodologies are the result of the long-time parallel history of the different trades. They established different vocabularies for their domains without looking across to neighbour trades. For a firm common Functional Requirements (FR) description, other description methods are necessary.

In the standard (IEC 61499, 2014) a function block model defined which is suitable for automation systems. It is based on the (IEC 61131, 2003) and defines function blocks with typed data points to bind different function blocks (see Figure 4 below):

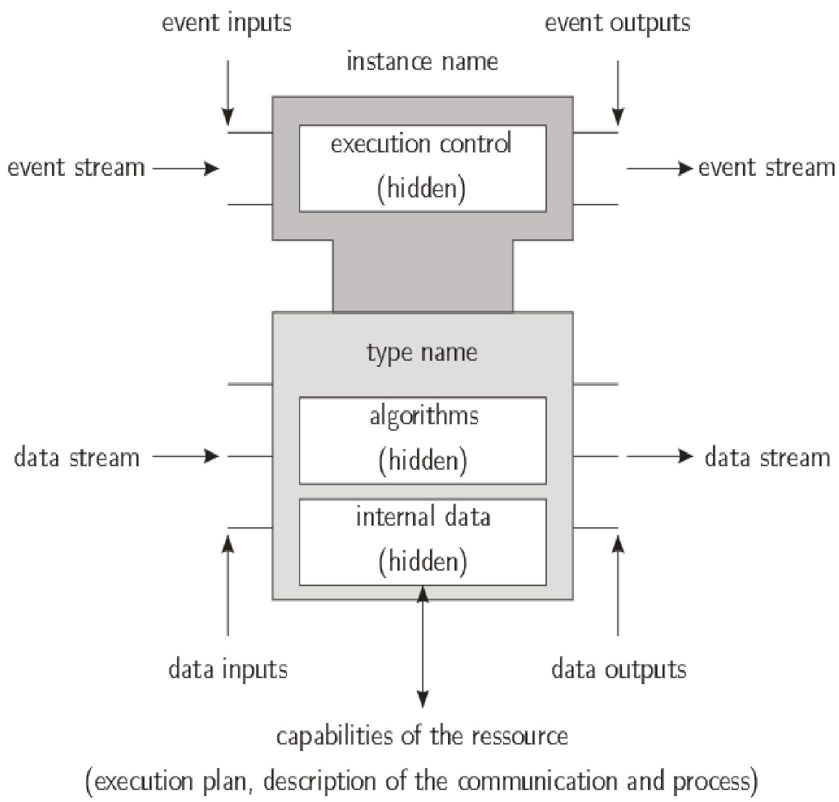

Figure 4: Function block from EN61499

(IEC 61499, 2014)

Furthermore, there is a set of German guidelines available or currently under development, which uses this function's block description to describe the functional aspects of building automation systems:

- VDI 3814: describes the general parts of the technical building systems like heating boilers, pumps etc. of building automation systems – this guideline was the basis of ISO16484 – the VDI 3814 is in the original form not function block based (only textual descriptions) but will currently completely rewritten with function blocks. The room automation functions, which are currently described in VDI 3813, will be integrated in this norm. (VDI3814:2017-07, 2017)
- VDI 3813: describes room automation functions for lighting, shading and HVAC within 53 functions – each function is described with an explaining text, with a unique graphic (the function block) and a table for explaining the data points of the function block. (VDI3813:2011-05, 2011)
- VDI 3812: describes the functions for AAL (Ambient Assisted Living) and additional needed trades in RAS also function block based – this guideline is still under construction. (VDI3812:2010, 2010). (TUD is working in the working groups for all of the three guidelines and leads the group for this standard development).

The main advantage of the function block based description is that on the one hand, it is possible to describe the functional requirements to the system and on the other hand, the functionality of the components or devices is describable in the same way. These guidelines and standards are currently the only standardized semantical description methods for room automation systems worldwide.

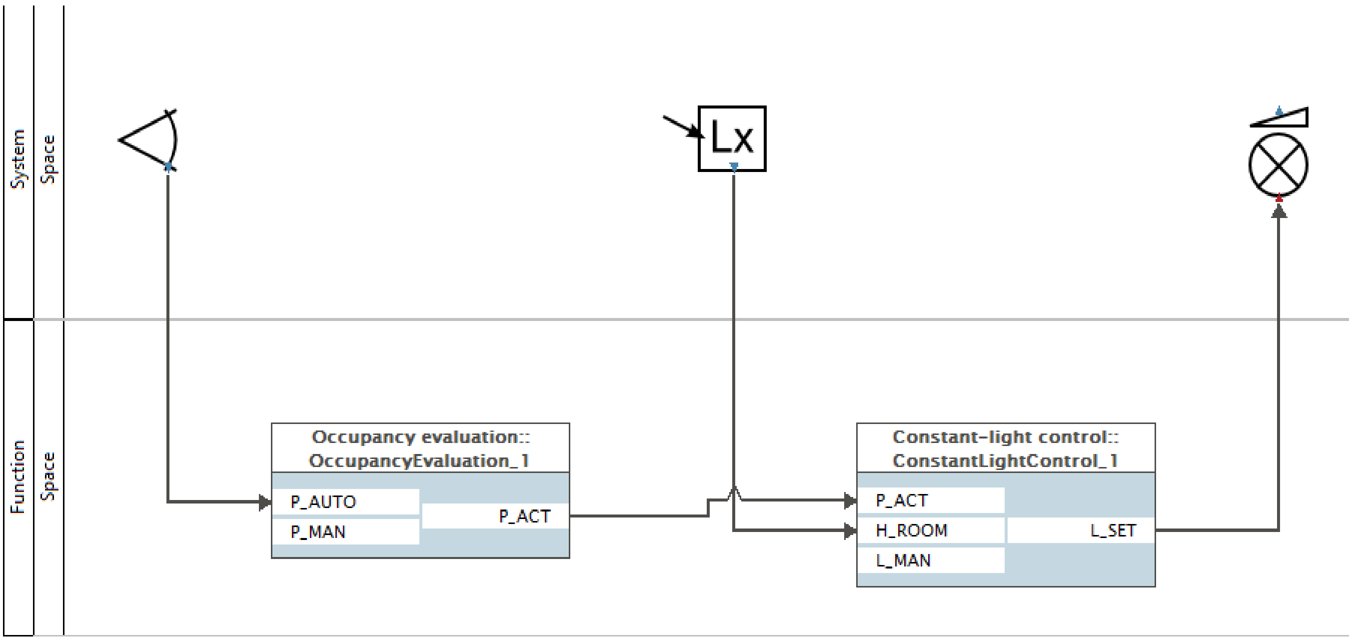

Figure 5: Function block based description of a constant light control system

In Figure 5 an example for a description of a constant light control is displayed. The upper symbols for sensors and actuators are also function blocks, but for an easier interpretation in AUTERAS (design tool of TUD for room automation systems) depicted with commonly known symbols. The lower two function blocks are describing control functions. The displayed form is the upper event part of the (IEC 61499, 2014) function blocks (see Figure 4). The lower part of function blocks in Figure 4 with the concrete data, this data is very specific and detailed, and it is not necessary to go deep in details in context of this report.

The Figure 6 overleaf shows an XML formulation of the constant light control system. In the upper part the used five function blocks are described and in the lower part the bindings between them.

The XML representation of the function block based system description can be used as an exchange format between different tools.

Originally, it was planned to use the well-known IFC-format (Industry Foundation Classes) in AUTERAS because they were developed especially for data exchange between stakeholders in the building sector. In addition, IFC provides capable models for building structures and the installed components.

However, a very important aspect is not describable with IFC: flows of events, data and information are not included in IFC w.r.t. (ISO 16739-1, 2018). Nevertheless, to describe the functionality of a building automation system this is a must-have. This is the reason, why a new XML scheme was introduced in AUTERAS to describe the functionalities and their connections together for a data exchange with other tools. The most important XML-fragments (like, the building structure or the devices) can be linked to corresponding parts in an IFC-based model.

```xml
<ga:functionalDescription>
    <ga:functionblocks>
        <ga:functionblock fbtype="luminance_sensor_indoor" name="Brightness measurement room_1"/>
        <ga:functionblock fbtype="constant_light_control" name="Constant-light control_1"/>
        <ga:functionblock fbtype="occupancy_control" name="Occupancy evaluation_1"/>
        <ga:functionblock fbtype="light_actuator_dimmable" name="Light actuator dimmable_1"/>
        <ga:functionblock fbtype="occupancy_sensor" name="Presence detection_1"/>
    </ga:functionblocks>
    <ga:connections>
        <ga:connection id="Brightness measurement room_1_H_ROOM_Constant-light control_1_H_ROOM">
            <ga:out port="H_ROOM" functionblock="Brightness measurement room_1"/>
            <ga:in port="H_ROOM" functionblock="Constant-light control_1"/>
        </ga:connection>
        <ga:connection id="Occupancy evaluation_1_P_ACT_Constant-light control_1_P_ACT">
            <ga:out port="P_ACT" functionblock="Occupancy evaluation_1"/>
            <ga:in port="P_ACT" functionblock="Constant-light control_1"/>
        </ga:connection>
        <ga:connection id="Presence detection_1_P_AUTO_Occupancy evaluation_1_P_AUTO">
            <ga:out port="P_AUTO" functionblock="Presence detection_1"/>
            <ga:in port="P_AUTO" functionblock="Occupancy evaluation_1"/>
        </ga:connection>
        <ga:connection id="Constant-light control_1_L_SET_Light actuator dimmable_1_L_SET">
            <ga:out port="L_SET" functionblock="Constant-light control_1"/>
            <ga:in port="L_SET" functionblock="Light actuator dimmable_1"/>
        </ga:connection>
    </ga:connections>
    <ga:inPorts/>
    <ga:outPorts/>
</ga:functionalDescription>
```

Figure 6: XML-description of a constant light control system

After the description of semantical functions the next section will provide a description methodology of functional requirements to a room automation system.

2.2 Requirement Feature Model of AUTERAS

2.2.1 Basic requirement model based on the VDI 3813

Based on the above mentioning possibility of using the manufacture- and technology-neutral function blocks from the (VDI3813:2011-05, 2011) to describe the functional requirements of BAS, a *model* to describe the system requirements was developed as part of the *requirement analysis* module of AUTERAS.

From all the 53 function blocks available in (VDI3813:2011-05, 2011), specific function blocks that are easily understandable for the common users (e.g. the owner of the BAS or the future occupants) and thus suitable to represent their wishes were chosen for the requirement model.

These include the function blocks, and also the corresponding requirements, for:

- Application functions: this type of requirements depicts the characteristics of the controlling of a specific trade and describes how a trade in the system should function (the wanted system behaviour);
- Operator and display functions: these requirements represent the possibilities for the direct interaction between the users and the BAS itself;
- Sensors and actuators: these requirements are often derived from the definition of the requirements for the application functions. Specific types of sensors and actuators are required to allow the realization of the full functionality of application functions. The actual number of these sensors and actuators depends on the scale of the system, which must be specified by the planner or the user as part of the requirements.

These requirements are defined for all the 3 trades in the (VDI3813:2011-05, 2011) for the control of light, shading and HVAC. Requirements that are related to more than one trade (e.g. a requirement for occupancy sensor can be defined as part of the requirement list for HVAC, Shading and Lighting) are grouped in a different neutral category. An overview of the requirements and their categorization can be seen in Figure 7.

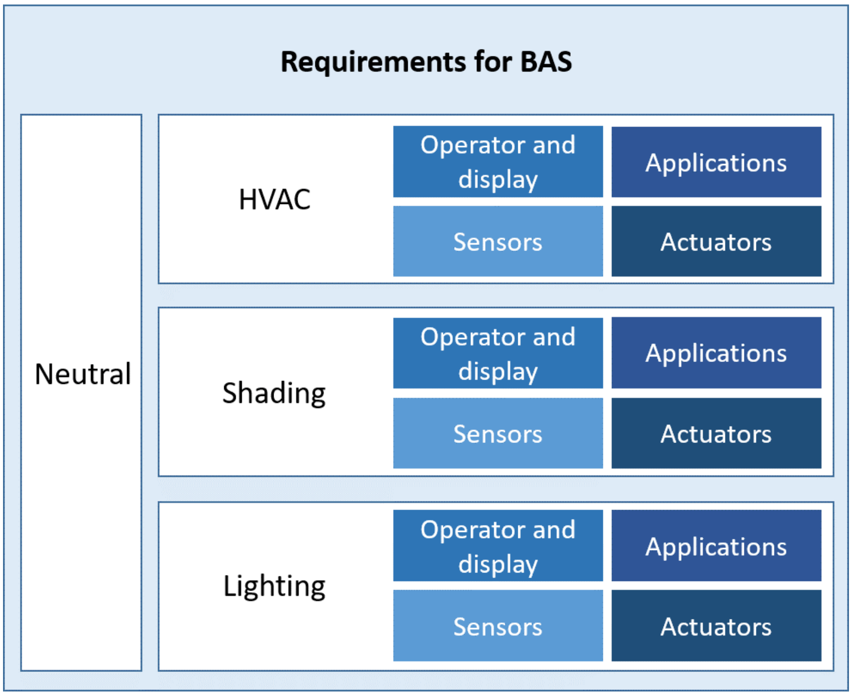

Figure 7 : Requirements for BAS

Along the requirement analysis process of the feature model (e.g. via interviews with planners) as a module of AUTERAS, it was determined that extensions of the wordings from (VDI3813:2011-05, 2011) were necessary to depict the common requirements for BAS. This includes the addition of specifications from general requirements. As an example, instead of the generic requirement for “light actuator”, it is required to choose between “dimmable light actuator” and “not-dimmable light actuator” as this decision affects the completeness and validity of the requirement for lighting control. Similarly, instead of the generic “actuate light” from type operator, it is necessary to specify between “switching function” and a “dimming function”. A list of extensions was determined and added to the original list of requirements.

2.2.2 Feature model/ Feature tree

The interface of the implemented requirement analysis is shown in Figure 8. The list of all requirements are organised as a feature model/feature tree. The hierarchy of the different nodes depicts the categorising of the requirements mentioned above. Each requirement is represented by a choosable feature. More generic features are added for the organisation purpose (e.g. the feature-nodes for the trades or for the categories of the specific features). The tool automatically selects features (explained in next section) which are dependent from another chosen feature. Also, impossible features will be deactivated (e.g. in case of the constant light is chosen, a non-dimmable light actuator is impossible). The user can define the multiplicity of the chosen feature (e.g. in case (s)he needs more than only one light actuator). If user interaction for some features is needed, the concerned features will be marked in colour red. The user gets also information about the validity state and energy efficiency class of his current selection.

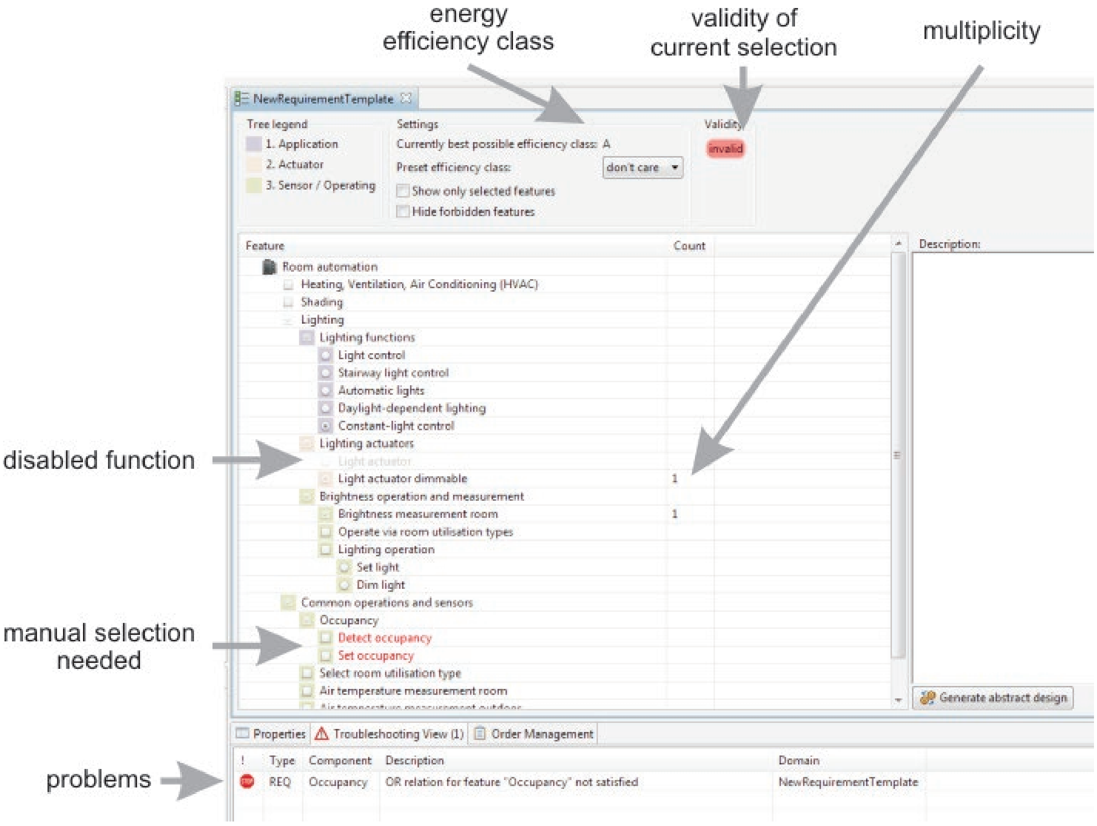

Figure 8: AUTERAS' requirement analysis editor

In order to support the users in understanding the different requirements, each feature in the feature tree was explained per clicked by the tool. Information about related requirement features are also shown, such as which further requirements are required when a specific requirement is selected.

2.2.3 Requirements as part of AUTERAS's user support

To support the tool's users during the requirement analysis, logical *relations* between features in the tree can be defined. In the current state of the requirement analysis module, a set of relations was already defined for the implemented VDI 3813 requirement features. This set of relations can be extended in the future as part of the extension of the module with requirements from other standards.

The first possibility is to integrate these relations directly into the feature tree. Relations between parent and child nodes as well as between sibling nodes can be defined.

This includes relationships such as:

- "*Mandatory*"-Relationship: the selection of a parent node leads to the selection of a specific child node
- "*Or*"-Relationship: the selection of a parent node requires the selection of at least one of the child nodes
- "*Alternative*"-Relationship: the selection of a parent node requires the selection of exactly one of the child nodes. As long as one child node is selected, its sibling nodes will not be selectable
- "*Optional*"-Relationship: the selection of a child node is optional, when the corresponding parent node is selected

Another possibility to define the relationship between features in the feature tree is to define rules in the form of logical *implication*. E.g.:

Constant light control → (Light sensor ∧ Dimmable light)

By using the implication as well as other logical operators like "and", "or", "negation" etc. it is possible to record different types of relationships between two or more requirement-features. This type of constraints can be defined for different features in the tree, independent from their relative positions to each other.

The relations and the implications that are defined as part of the expert knowledge for the requirement analysis, in which the tool developers implement the common knowledge in the design of BAS into the tool. The basis for the definition of relationship and the implication was mostly derived from the VDI guidelines mentioned in chapter 2 of this report. During the selection of requirements by the user, an implemented SAT-Solver continuously check the current list of the selected requirement features and compare this with the predefined relations and implications.

Originally it was planned to describe the requirement features within an ontology and use a reasoner for determining the validity of the set of selected features (see task T6.1 description). This was rudimentary implemented, but later on replaced by the described solution with simple relations and implications between different requirement features and using a SAT-solver (sat4j [Sat]). The new solution has several advantages. At the moment are the functional requirements based on the (VDI3813:2011-05, 2011) integrated in the requirement analysis tool. If a later user wants to extend the requirement features, (s)he does not need to be an expert in ontology domain. But the major reason was the performance of the available reasoners. Several free reasoners were tested (Pellet [Pel], HermiT [Her], FaCT++ [FaC]). Pellet is admittedly the most known reasoner (it's the standard reasoner of the tool Protegé), but a bit slow. Also FaCT++ as the fastest tested reasoner has a lower performance for this use case than the SAT-solver solution realized in the end. A very good performance is needed, because the checking of validity of selected features must be performed after each user interaction. Another reason for switching the methodology was that the latest updates for the tested reasoners are very old and so the support in the future is unclear. (Information System Group, 2019), (The University of Manchester, 2019), (W3C, 2019).

Already in the implementation phase some effects were shown up. The definition of new relations and implications is very error-prone. If a later user of the tool wishes to extend the requirements (s)he must be an expert in ontology technologies to describe the new requirements. This is contrary to the intended flexibility and extensibility of the tool. In the evaluation of the solution against the description with relations and implications and using a SAT-solver it was also shown that the ontology-based variant with using a reasoner has a much worse performance. Because of that after each user interaction the complete feature tree will be checked, the performance issue becomes significant. Because of these disadvantages of using ontology-based technologies in this special field, the other methodology was integrated in the requirement analysis tool.

As a result, suggestions including inclusion and exclusion of requirement features in the feature tree will be automatically derived and presented to the user. With this tool's feature for user support, the process of requirement analysis using AUTERAS will be faster, more efficient and less prone to error.

2.2.4 Requirement template and open format for the requirement model

The selected requirements are grouped into a requirement template, which represents all requirements defined by a user. The usage of the requirement templates depends on the user: one template can represent the requirements specified for a specific trade in a room in BAS or the requirements defined for a room type (more details in chapter 4.2 of this report).

For data storage and exchanging the requirement templates between different, a description in XML was chosen as the open format for the requirement templates. An example of the description can be found in Figure 9 below:


```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ga:requirementtemplate name="ConstantLightRequirement" xmlns:gaext="http://gaentwurf.de/model/extensions" xmlns:gafl=
"http://gaentwurf.de/model/floor" xmlns:ga="http://gaentwurf.de/model">
    <ga:configuration ordernumber="0" id="ConstantLightRequirement_0">
        <ga:childconfigurations/>
        <ga:requiredfeatures>
            <ga:requiredfeature feature="LuminanceSensor" count="1" derived="true" userselected="false" id=
            "ConstantLightRequirement_0__LuminanceSensor"/>
            <ga:requiredfeature feature="Lighting" count="1" derived="true" userselected="false" id=
            "ConstantLightRequirement_0__Lighting"/>
            <ga:requiredfeature feature="OccupancySensor" count="1" derived="false" userselected="true" id=
            "ConstantLightRequirement_0__OccupancySensor"/>
            <ga:requiredfeature feature="RoomAutomation" count="1" derived="false" userselected="true" id=
            "ConstantLightRequirement_0__RoomAutomation"/>
            <ga:requiredfeature feature="Common" count="1" derived="true" userselected="false" id=
            "ConstantLightRequirement_0__Common"/>
            <ga:requiredfeature feature="Occupancy" count="1" derived="true" userselected="false" id=
            "ConstantLightRequirement_0__Occupancy"/>
            <ga:requiredfeature feature="LightingActuators" count="1" derived="true" userselected="false" id=
            "ConstantLightRequirement_0__LightingActuators"/>
            <ga:requiredfeature feature="LightActuator_dimmable" count="1" derived="true" userselected="false" id=
            "ConstantLightRequirement_0__LightActuator_dimmable"/>
            <ga:requiredfeature feature="OccupancySwitch" count="1" derived="false" userselected="true" id=
            "ConstantLightRequirement_0__OccupancySwitch"/>
            <ga:requiredfeature feature="ConstantLight" count="1" derived="false" userselected="true" id=
            "ConstantLightRequirement_0__ConstantLight"/>
            <ga:requiredfeature feature="LightingOperAndSensor" count="1" derived="true" userselected="false" id=
            "ConstantLightRequirement_0__LightingOperAndSensor"/>
            <ga:requiredfeature feature="LightingFunctions" count="1" derived="true" userselected="false" id=
            "ConstantLightRequirement_0__LightingFunctions"/>
        </ga:requiredfeatures>
    </ga:configuration>
    <ga:segmenttemplates/>
</ga:requirementtemplate>
```


Figure 9: XML Description for a requirement template

The example shows the list of requirement features needed for a constant light control as part of a requirement template. The XML-description contains information about

- Which features in the feature tree were selected
- The multiplicity of the selected features
- If the selection was done by the user or if the selection was automatically done by the SAT-solver

As well as further information that is required to reconstruct the represented state of the feature tree.

3 Non-Functional Requirements

Firstly in this chapter it is primarily based on the extension of AUTERAS (Technische Universität Dresden, 2013) in BIM4EEB regarding common non-functional requirements will be described. In the second section the integration of a special non-functional requirement, the energy efficiency classes of RAS, into AUTERAS as well as the problems by using different standards and norms will be described.

3.1 Description method for non-functional requirements in AUTERAS

In general a Non-Functional Requirement (NFR) can be formed from nearly every property of a component except the core functionality. As an example there is a light actuator device. It has two properties: a price of 10€ and the color of the device is red. A NFR could now be defined with the color or the price. For example a user can define that he wants only red devices or green devices or devices cheaper than €20. But (s)he is not able to define a NFR like the dimensions of the device should be smaller than 10x10x10cm, because the available device does not have a property like that.
In practice there are a lot of different components with very different properties available on the market. For instance in the small domain of room automation, there are thousands of devices available, and each manufacturer describes his products in a different way. But no standard, guideline or regulation exists, to describe the properties of products or which properties must be described.
Summarising, the proposal is to search for all properties of the available components in the considered component repository. To achieve this, there are two options:

1. Only properties which are defined for all devices are allowed to be used as requirements;
2. All properties can be used for defining requirements. In this case there is the difficulty of how to evaluate components which do not provide a specific property. For such situations a default value can be used or a better solution would be to say this device could be used to fulfill the requirement but mark it as “unsure” and the user has to check the validity manually afterwards.

In the current implementation of AUTERAS’ device search (this will be explained in more details in the followig WP6 report D6.2) the first option is realized. The reason was that the second option increases the complexity of the device search’s solution space enormously, but the goal was to reduce the complexity as much as possible.
The properties of a component can be directly linked to the component itself or a link to a description data base (like eCl@ss-data bases or information available in common IFC-based descriptions) can be attached.
The NFRs can be very simple like “all devices must be red” or can be combined with methods of sentential logic to much more complex NFR like “all devices must be red or must be flush-mounted installable”.
The mostly available properties of components in the domain of room automation are:

- manufacturer of the device;
- platform (e.g. LON, KNX, enOcean);
- price;
- ordernumber;
- ingress protection;
- communication medium;
- power supply;
- possible installation location.

But for a lot of devices is much more information available, e.g. form, size, color, information about contact points (material, size, location,…) etc.

A maschine-readable and –evaluable NFR has the form of an arithmetic term:

property operator comparative value

The property can be any property of a device. The operator can be any relational operator (<, <=, >, >=, =, ≠). Attention should be paid that not all operators are meaningful for all kinds of properties. So in case of properties that are string-properties only the equality or inequality are useful.
For properties that have numbers as values, all operators can be used. The comparative value must have the same data type as the property itself.
Following are some simple examples for NFRs to devices displayed:

devices color = “red”
devices price < 100€
devices manufacturer = “Semiens”

Like introduced above, these simple NFRs can be combined to more complex NFRs with the logical operators ∧, ∨ and ! (not).

((devices price < 100€) ∨ (devices manufacturer = “Semiens”)) ∧ (!((devices price < 100€) ∧ (devices manufacturer = “Semiens”)))

This example describes that the devices should be cheaper than 100€ or in case it is not possible the manufacturer of the devices should be “Semiens”.
But there are also other types of NFRs. The most important is the coherence between the devices and the building in which the devices should be installed. For example the materials of the building structures should be taken into account. So it is not very useful to try to install a screw-on light switch on a glass wall.
Also a lot of regularities define where devices should be installed and where it is forbidden. In the German standard (DIN 18040, 2014) it is regulated that a light switch should be installed in a height of 105cm over the floor and in case it is a barrier-free building in a height of 85cm. Such regularities must be taken into account not only in cases of new buildings but also by retrofitting buildings. Another interesting example are smoke alarm devices. There exists several regularities like DIN 14675 (withdrawn now) replaced by the (DIN 14676-1:2018-12, 2018) on how to install them in the rooms (non-functional requirements). But for the same devices there are also regularities regarding their functionality (functionality) like (DIN EN 14604:2009-02, 2009). The problem of this functions is that they are not semantically described like the other room functions in (VDI3813:2011-05, 2011). This shortcoming should be corrected by the current work in (VDI3814:2017-07, 2017).

3.2 Targeted systems energy classes as an example of complex non-functional requirements

As stated in the section above it is possible to define non-functional requirements to the single functional requirements for a room type. For most non-functional requirements (like color, price, energy consumption of realizing devices) it is suitable. But there are also non-functional requirements to an system, which influence directly the functional aspects. A very important example of this phenomenon is the Energy Efficiency Class (EEC) which a system can fulfill. The energy efficiency or performance of a room automation system describes the relation of the benefit of the system and the needed energy quantitatively. In the European norm (EN 15232:2012, 2012) are four energy efficiency classes introduced:

- **Class D**: BAS which are not energy efficient. Buildings which are classified in this class should be modernized. In case of a new building should be certified, its BAS must have a higher class than this one.
- **Class C**: this class is for “standard” BAS.
- **Class B**: the BAS has advanced functions.
- **Class A**: this class is for highly energy-efficient BAS.

To reach a higher class, all functions which are needed for the next lower class and some additional functions must be fulfilled. The norm defines the EEC for residential as well as for non-residential buildings. In Table 3 is an example for such a classification depicted. Like stated a simpler equipped building with only a constant temperature control has a lower EEC than a building where the heat control is dependent on the outside temperature (which influences also the inner building temperature) or the load of the system.

Table 3: Example of a BAS classification in EN15232

Classification			**definition of the classes**							
			residential				**non-residential**			
			D	**C**	**B**	**A**	**D**	**C**	**B**	**A**
1.6	Controlling of the heat-generator (incinerator or district heating)									
	0	constant temperature control	x				x			
	1	variable temperature control depending from outside temperature	x	x			x	x		
	2	load-depended variable temperature control	x	x	x	x	x	x	x	x

In the (EN 15232:2012, 2012) there are a lot of functions for different trades like HVAC, shading and lighting available.

The (VDI3813:2011-05, 2011) has taken this classification and has also defined these EECs for room automation functions. But there exist some problems in both standards. One minor issue is that VDI3813 focuses mainly on non-residential buildings. This is the reason that in VDI3813 only EECs for non-residential buildings are given. But the main issue is that some of the room functions defined in VDI3813 have no EEC definition in both standards. In the following table these functions are listed with a proposal how they should be classified. The proposals were directly implemented in the current version of AUTERAS.

Table 4: Functions without EEC in VDI 3813 and EN 15232

Functions			**proposal**							
			residential				**non-residential**			
function	trade	function description	**D**	**C**	**B**	**A**	**D**	**C**	**B**	**A**
weather protection	shading	The function weather protection prevents damage to external sunshading equipment due to wind, rain or icing. Whereas wind velocity and rain are measured directly by means of sensors, the icing hazard is predicted indirectly through evaluation of precipitation in combination with the outdoor temperature.	x	x	x	x	x	x	x	x
Automatic twilight control	shading	The function automatic twilight control allows sunshading equipment to be positioned in accordance with outdoor brightness. For instance, it allows the closing of the sunshade during night hours in order to, reduce cooling down via the windows or reduce light emission by the building.	x	x	x	x	x	x	x	x

It would be beneficial to include examples illustrating different levels of control rather than all classes indicated, but these are the lacks in those standards under development at the moment. There are also functions in VDI 3813 which got an EEC there, but there is no corresponding element in (EN

15232:2012, 2012).

Table 5: Functions with EEC-definition only in VDI 3813

Functions			**proposal**							
			residential				**non-residential**			
function	trade	function description	**D**	**C**	**B**	**A**	**D**	**C**	**B**	**A**
Automatic solar control	shading	The Automatic solar control prevents user discomfort due to exposure to high-intensity sunrays by moving the sunshade to a defined fixed anti-glare position as soon as the daylight exceeds a defined illuminance. With fading brightness, the sunshade is moved to a parking position.	x	x			x	x		

Another minor issue is that in (EN 15232:2012, 2012) the definition for systems with presence detection is a bit to strict. In case of manual presence detection (e.g. with a card or switch) the maximum EEC of the system is C. Also if the system consists only from much higher functions (except of the manual presence detection) the maximum EEC for the whole system is B. Here it will be proposed to give only a smaller penalty of one EEC lower than with automatic presence detection. When the planner defines the needed functions for the BAS, the AUTERAS calculates directly which EEC can be reached. Another possiblility is that the user firstly defines which EEC he want to achieve and the planning system will inform him, what is needed to achieve this Energy Efficiency Class (EEC).

4 An open format for system specification

For the communication between the different parts, aspects, tools and models of BIM4EEB a common system model is needed. For providing the functional and non-functional aspects of RAS that are defined within AUTERASplan/design the existing AUTERASstore was enhanced with a web-based SOAP-interface.

In the beginning of this chapter AUTERAS' device model will be explained, as it is the basis of the system model. Secondly, in this chapter the concept of AUTERAS' template mechanism is explained. This is needed, because of all functional and non-functional aspects are defined in such templates to reduce the effort in the planning process of RAS enormously. After it, the system model, based on AUTERASstore, will be explained as well as the provided functionalities.

4.1 Device description

The description method for devices for building automation systems must consider all aspects of a device. Because of the described weakness of the IFC-specification regarding functional aspects a XML-based format is proposed to describe components.
A device description consists in general of two parts:

1. functional description
   The functional aspects should described in the way indicated above with function blocks and connections between them. In case of room automation systems the set of possible functions is defined in VDI 3813. For other aspects in BAS this function set is under construction in VDI 3814. But it is also possible to define our own function set.
2. non-functional description
   Non-functional aspects of devices could be added as simple key-value pairs to the devices.

In Figure 10 is an example of such a device description in XML displayed. The functional description is inside the XML-tag "functionalDescription" while the non-functional description is inside the XML-tag "properties". The functional description consists of four different parts:

1. "functionblocks"
   The function blocks which are listed inside this tag are the VDI3813 function blocks which are realized by this device. Important to say is that more than only one function block of the same type can be implemented by a device. Only the names of the function blocks have to be unique for identification. A practical example is a multiple light actuator which implements for example four light actuators.
2. "connections"
   The connections are directed bindings from an output data point of a function block or an input data point of the device to an input data point of a function block or an output data point of the device. Interesting is that it is also possible to connect an output data point with an input data point of the same function to realize feedbacks. But feedback loops are mostly over two or more functions. These feedbacks are often used to monitor the actual values of actuators or to affect controllers with the actual actuator state.
   The definition of a connection consists of a unique name and a pair of its source and destination.
3. "inPorts"
   "inPorts" are the input data points of the device.
4. "outPorts"
   "outPorts" are the output data points of the device.

The examplary device below has no in- and output data points, so it can work only with its own without any communication with other devices.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ga:device name="TestDevice" xmlns:ga="http://gaentwurf.de/model">
    <ga:functionalDescription>
        <ga:functionblocks>
            <ga:functionblock fbtype="luminance_sensor_indoor" name="Brightness measurement room_1"/>
            <ga:functionblock fbtype="constant_light_control" name="Constant-light control_1"/>
            <ga:functionblock fbtype="occupancy_control" name="Occupancy evaluation_1"/>
            <ga:functionblock fbtype="light_actuator_dimmable" name="Light actuator dimmable_1"/>
            <ga:functionblock fbtype="occupancy_sensor" name="Presence detection_1"/>
        </ga:functionblocks>
        <ga:connections>
            <ga:connection id="Brightness measurement room_1_H_ROOM_Constant-light control_1_H_ROOM">
                <ga:out port="H_ROOM" functionblock="Brightness measurement room_1"/>
                <ga:in port="H_ROOM" functionblock="Constant-light control_1"/>
            </ga:connection>
            <ga:connection id="Occupancy evaluation_1_P_ACT_Constant-light control_1_P_ACT">
                <ga:out port="P_ACT" functionblock="Occupancy evaluation_1"/>
                <ga:in port="P_ACT" functionblock="Constant-light control_1"/>
            </ga:connection>
            <ga:connection id="Presence detection_1_P_AUTO_Occupancy evaluation_1_P_AUTO">
                <ga:out port="P_AUTO" functionblock="Presence detection_1"/>
                <ga:in port="P_AUTO" functionblock="Occupancy evaluation_1"/>
            </ga:connection>
            <ga:connection id="Constant-light control_1_L_SET_Light actuator dimmable_1_L_SET">
                <ga:out port="L_SET" functionblock="Constant-light control_1"/>
                <ga:in port="L_SET" functionblock="Light actuator dimmable_1"/>
            </ga:connection>
        </ga:connections>
        <ga:inPorts/>
        <ga:outPorts/>
    </ga:functionalDescription>
    <ga:properties>
        <ga:property type="manufacturer">Semiens</ga:property>
        <ga:property type="price">567.89</ga:property>
        <ga:property type="platform">LON</ga:property>
        <ga:property type="orderNumber">123-456/789</ga:property>
        <ga:property type="protectionClass">IP65</ga:property>
        <ga:property type="communicationMedium">twisted pair</ga:property>
        <ga:property type="powerSupply">230V</ga:property>
        <ga:property type="installationLocation">wall</ga:property>
        <ga:property type="installationForm">cap rail</ga:property>
    </ga:properties>
</ga:device>
```

Figure 10: Example of a device description

As presented in the previous sections of this report, the description of functional requirements is strongly similar to the description of devices functionalities.

In the second part of the device description are the properties of the device listed. Such a property consists only of the property type and its value. Over these properties can now the non-functional requirements be set up.

4.2 Room templates

An automated room consists of the room with its information itself as well as automation devices for room automation functionality. For the room information and also general information about the automation devices the models described in D3.x should be used. To combine them with functionality description another model must be added (because of the missing semantic functional description methods inside the other models like IFC, see chapter 3 of this report).
In the (VDI3813:2011-05, 2011) not only the functionality of room automation is described, but it also contains a small shell model for building structures, from property portfolio, over property, building, area, room down to segment. This model contains the most important structures. In WP6 this model was the basis. But it had to be enhanced with structural components like facades. The reason was mainly to allow the connection to the other models for buildings' descriptions in the BIM4EEB WP3 activities.

Important to say is that there is no need for any other building information except of its general structure, to describe its functionality. This resulted in the solution to separate this functionality decription from the other building models.

Each of the building structures has now description of the contained room automation functionality (see chapter 3 of this report). It is recommended to describe not the single rooms, but room types. For example in larger office buildings there are a lot of similar equipped rooms or in a larger apartment building there are a lot of similar flats. A room type has to be described only one time and can be linked to each room of the same type. For the functional description of the automation systems for the different room types it is recommended to use the same description method with (VDI3813:2011-05, 2011) function blocks as described above for the device descriptions. A small example for a functional description of a room type is depicted in Figure 5. An advantage of this methodology is that it is easily possible to define also non-functional requirements to each of the function blocks in a room template. For example it can be demanded that *"all devices for sensors and actuators must have the same color AND all control functions must be flush-mountable realized"*.

If the templates will be assigned to different locations, they become template instances (see Figure 11 and Figure 12).

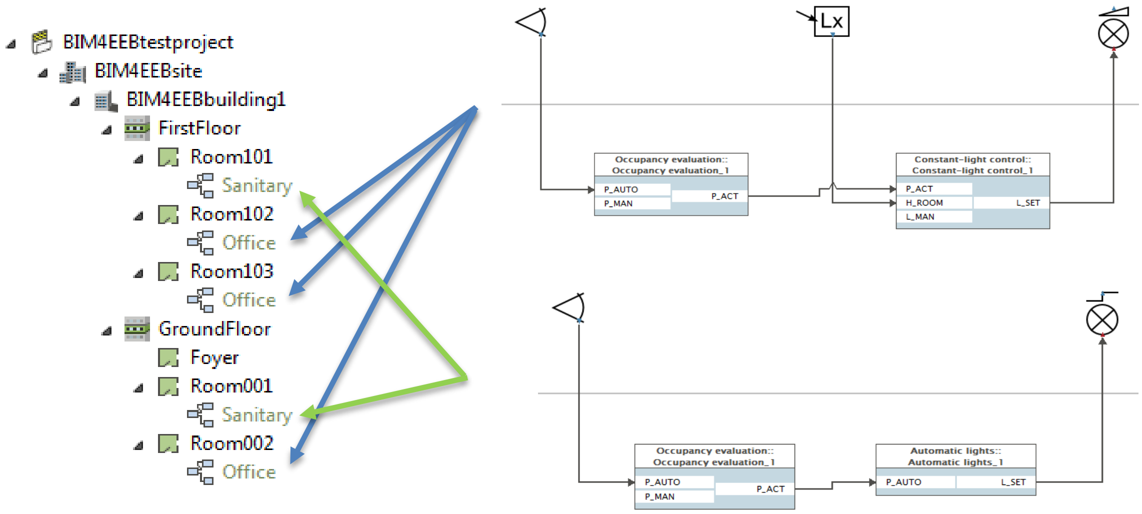

Figure 11: Example for building structure with functional requirements for different room types

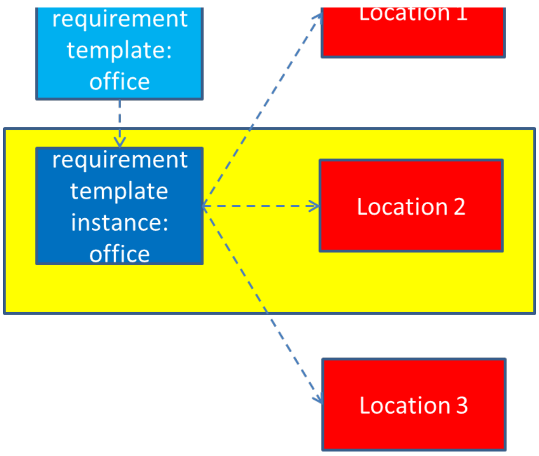

Figure 12: From requirement template to requirement template instance

For a better depiction in AUTERASplan different template types were established:

- requirement templates;

  These templates contain the selected requirement features from the requirement analysis (see section 2.2 of this report);

- abstract templates;

  These templates were automatically generated from AUTERSAplan and can be enhanced with additional functional and non-functional requirements. They will be stored also as XML similar to the example in Figure 6 enhanced with the non-functional requirements as properties.

  The abstract templates describe the planned BAS formal. They are appropriate for using them in the procurement process because they are platform- and vendor-independent. Also a cost estimation for the BAS is possible.

The third template type of AUTERAS is:

- concrete templates

  These templates contain concrete devices which fulfill the abstract templates. In the following D6.2 this template type will be explained more in detail. They will be generetad by AUTERASdesign.

4.3 System model and getting information

For interacting of the RAS-model with the other components and models of BIM4EEB, especially the models described in the BIM4EEB WP3 reports, an open format and infrastructure is needed.

The tool AUTERASstore from TUD-TIS is a good basis for this task. It is a store for AUTERAS-RAS-projects. The planning tools AUTERASplan/store can directly load and save projects from and to AUTERASstore (see Figure 13).

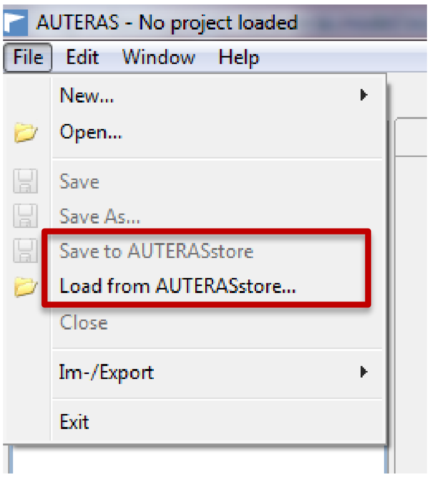

Figure 13: Extension of AUTERASplan/design for AUTERASstore

AUTERASstore was now extended with a SOAP-based and a RESTful web-service interface. AUTERASstore itself is hosted by (Technische Universität Dresden, 2013).

Its SOAP-based web-service interface is described with a WSDL-interface and available under:

http://141.76.83.178:9093/bim4eeb_soap?wsdl

The RESTful webservice was extended for better testing and using with a swagger [Swa] interface. This interface is available under:

http://141.76.83.178:9092/BIM4EEB_REST/

The swagger-API provides additionally to the RESTful web service interface of AUTERAS a browser-based front-end (see Figure 14). All the methods of the web service are listed and additional information, like parameters or the request URL, are displayed. A very interesting feature is the possibility to test the method call directly from this web page and getting back the result the result.

Figure 14: RESTful swagger interface of AUTERASstore

AUTERASstore now provides both interfaces (SOAP and REST) and many functions to get any information or to change information in case of system changes. Following the most important functions will be explained more in detail.

Table 6 : General methods of AUTERASstore

Method name	Parameters	Description
getProjectNames	String:projectOwner	to get the names (ID) of all projects stored for the given projectOwner
getProject	String:projectOwner String:projectName	returns the complete AUTERAS-project with the name projectName as XML

There are also hidden methods, like “addProject” or “deleteProject” to add new projects into AUTERAS or delelte them. To ensure the consistency of the projects these functions should be used only by AUTERASplan/design. The method “getProject” gives the whole project including all templates and their instances into the single locations. They contain the requirement templates and the functional templates with the defined functional and non-functional requirements, as well as the device templates with the installed RAS-devices in the dedicated rooms as well as their connection-schemes (see Figure 15 and Figure 16).

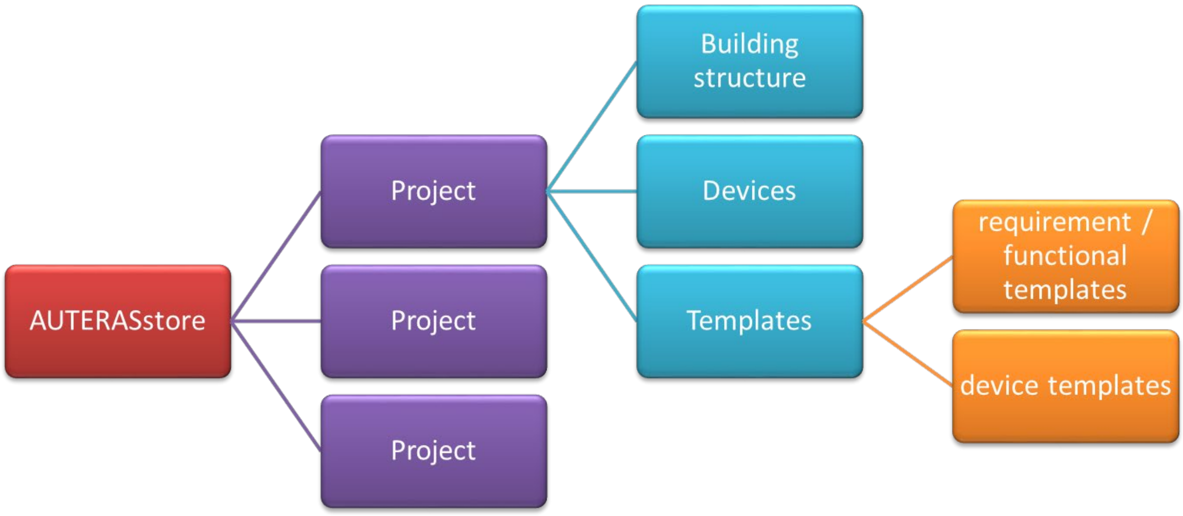

Figure 15: Project structure in AUTERASstore

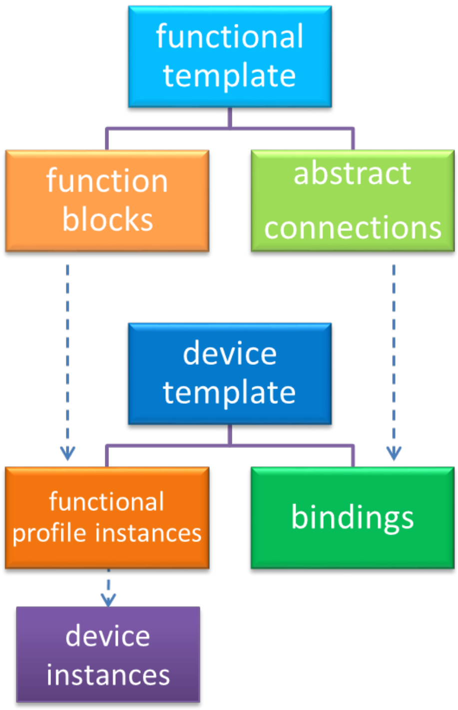

Figure 16: Mapping between functional and device templates

Table 7 : Location related methods of AUTERASstore

Method name	Parameters	Description
getSubLocations	String:projectOwner String:projectName String:parentLocationName	returns the names of all sub-locations of parentLocationName, in case there are no sub-locations an empty set will be returned.
getParentLocation	String:projectOwner String:projectName String:locationName	returns the parent location name of the given locationName, the root parent location is always “Project”, it has no parent location.
getAllLocationNames	String:projectOwner String:projectName	returns all location names of the project (without the structure)
getLocationByName	String:projectOwner String:projectName String:locationName	returns the complete location object for the given locationName with all its properties, templates (requirements, functional, devices), installed devices, data points, realized functions
getLocationPropertyList	String:projectOwner String:projectName String:locationName	returns all property names for the given locationName
getLocationPropertyValue	String:projectOwner String:projectName String:locationName String:propertyKey	returns the property value for the given propertyKey of locationName
getLocationsByPropertyKey	String:projectOwner String:projectName String:propertyKey	returns all location names which have an property defined with propertyKey as name
getLocationsByPropertyValue	String:projectOwner String:projectName String:propertyKey String:propertyValue	returns all location names which have an property defined with propertyKey and same propertyValue
setLocationProperty	String:projectOwner String:projectName String:locationName String:propertyKey String:propertyValue	adds or changes a property to a location

Table 8 : Functional RAS-element methods of AUTERASstore

Method name	Description
getLocationDeviceNames	gets the names of all devices in a location
getDeviceFunctionNames	gets the names of all functions from a dedicated device
getFunctionIn/OutDatapointNames	gets the names of all data points from a dedicated function of a device
getDevicePropertyList	gets all property names (the property keys) defined to a device
getDevicePropertyValue	gets the property value defined to a device propertyKey
setDeviceProperty	adds or changes a property value to/of a device

The location and element properties can be used also for providing links to the corresponding elements in other models and stores like the BIM4EEB-ontology (see WP3 reports) by their element IDs. For a bi-directional communication, it is recommended to store in the other models' links to the corresponding elements in AUTERASstore. For doing this the unique element names can be used.

The functional and non-functional requirements to a location or an RAS-element in it are part of the templates defined in the locations. They can be gotten with the method "getLocationByName" (see Table 7). The list of functions is not being fixed; they can easily be enhanced if needed in the project life time.

The functions can be easily invoked by a tool for web service calls, like SoapUI which is a free tool (GNU LGPL) in its basis version or by implementing the web service interface in any higher programming language e.g. SOAP (Simple Object Access Protocol, (TechTarget, 2019)) as protocol is platform-independent (e.g. generate most of the access code with wsdl2java from Apache Axis2 framework in case of using JAVA with the given open WSDL-file).

5 Conclusions

This deliverable is mainly focused on planning methodologies of BAS in the pre-procurement phase. As it was previously mentioned, the common formats for building description (e.g. IFC (ISO 16739-1, 2018)) are not fully suitable to describe those BAS functionalities as well as their logical interconnections, so the approach based on the German guideline (VDI3813:2011-05, 2011) was introduced. For the planning process of BAS for new buildings, as well for renovation of existing buildings, the planning tool AUTERAS (Technische Universität Dresden, 2013) will be used in this BIM4EEB research project.

This tool has formerly planned only functionalities of Room Automation Systems. It is advanced to handle non-functional requirements now. Additionally it is now able to handle coarser functional requirements. This simplifies the usage especially for inexperienced users like the occupants of the buildings. All the planning results will be stored in a system model store for Room Automation Systems descriptions which is web service based and has two interfaces for access via SOAP (exchange format: XML) and REST (exchange format: JSON).

The stored data can be easily linked to other models in other stores, so an integrated working process within different tools and mentalities is ensured. Special attention was taken on aspects of using available standards and guidelines as much as possible to prevent proprietary implementations and formats. Particularly for the interchange format it was not possible (mentioned above), but the explained new format was designed small, easy to use and linkable to other descriptions. Another important goal was the flexibility for new requirements and functionalities. All the components were prepared that a user can easily integrate new components.

In summary, the key achievements mentioned in this deliverable are relevant for further use in our BIM4EEB research activities, i.e.:

1. Analysis of the planning process of Building Automation Systems wih use of planning tool AUTERAS from (Technische Universität Dresden, 2013);
2. Development of an XML-based description for room functions based on well-established standards and guidelines like (ISO 16484-2:2004, 2004) or (VDI3813:2011-05, 2011);
3. Researched methodology for describing functional requirements on Room Automation Systems as well as development and implementation of a user-oriented Requirement Editor;
4. Developed methodology to enrich functional requirements with non-functional requirements as well as an analysis and integration of energy efficiency classes into the Requirement Editor;
5. Development and implementation of the web-based System Model for Room Automation Systems in building's structures with a SOAP- and a RESTful-interface.

The tool Auteras was enhanced on some of its components to cope with BIM4EEB WP6-relevant requirements, i.e.:

- AUTERASmodel: The component model of the whole AUTERAS tool suite was enhanced for describing non-functional requirements;
- AUTERASplan: The requirement definition editor was enhanced to define non-functional requirements like the energy efficiency class of the Room Automation System;
- AUTERASstore: The store of Room Automation System definitions was enhanced with new interfaces (SOAP and REST) to allow the interaction of the results of the planning with AUTERAS with other tools.

The newly defined non-functional requirements will be further defined in T6.2 report. This task focuses on the search for devices fulfilling the defined requirements.
To finally summarise, the work being done in this Task 6.1 (D6.1 V2.0 updated report) is provided good basis for the following WP6 tasks, where results of the pre-procurement planning activities were further used. This report is reviewed based on comments received from the BIM4EEB project officer and monitor.

6 Bibliography

BS EN 16310, 2013. *Engineering services Terminology to describe engineering services for building, infrastructure and industrial facilities.* s.l.:BSI Standard Limited.

DIN 14676-1:2018-12, 2018. *Smoke alarm devices for use in residential buildings, apartments and rooms with similar purposes,* Germany: Beuth.

DIN 18040, 2014. *Barrierefreies Bauen (Construction of accessible buildings).* [Online] Available at: https://ec.europa.eu/eip/ageing/standards/home/accessibility-and-design-all/din-18040_en

DIN EN 14604:2009-02, 2009. *Smoke alarm devices; German version EN 14604:2005,* Germany: Beuth.

eCl@ss, 2019. *The eCl@ss Standard.* [Online] Available at: https://www.eclass.eu/en/standard.html

EN 15232:2012, 2012. *Energy performance of buildings -Impact of Building Automation, Controls and Building Management,* s.l.: BUILD UP.

Gatta, O., 2017. Procurement process and service delivery in the UN organisation. International Journal of Supply Chain and Logistics, 1(2), pp. 82-112.. *International Journal of Supply Chain Management,* 2(1), pp. 57-79.

Gordon, S., 2009. *Supplier evaluation and performance exellence. A Guide to meaningful metrics and successful results.* 1 ed. NY: J. Ross Publishers.

HOAI:2013-07, 2013. *Honorarordnung für Architekten und Ingenieure - Official Scale of Fees for services by Architects and Engineers. Bundesrepublik Deutschland..* [Online] Available at: https://www.gesetze-im-internet.de/hoai_2013/HOAI.pdf

IEC 61131, 2003. *IEC standard for programmable controllers,* France: International Electrotechnical Commission.

IEC 61499, 2014. *The New Standard in Automation,* France: International Electrotechnical Commission.

Information System Group, 2019. *HermiT OWL Reasoner.* [Online] Available at: http://www.hermit-reasoner.com/

International Organization for Standardization, 2013. *Standards catalogue, ISO16739:2013.* [Online] Available at: https://www.iso.org/standard/51622.html

Ishola, T., 2010. *Procurement and supply management,* Victoria Island, Lagos: s.n.

ISO 16484-2:2004, 2004. *International Organisation for Standardisation ISO 16484-2:2004(en), Building automation and control systems (BACS) — Part 2: Hardware.* [Online] Available at: https://www.iso.org/obp/ui/#iso:std:iso:16484:-2:ed-1:v1:en

ISO 16739-1, 2018. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries - Part 1: Data schema,* s.l.: ISO.

ISO 16757-1:2015, 2015. *Data structures for electronic product catalogues for building services — Part 1: Concepts, architecture and model.* [Online] Available at: https://www.iso.org/standard/57613.html

Manu, C., 2005. *The Principle Of Value For Money In Procurement,* Ghana: GhanaWeb.

Matechak, J., 2009. *Fighting Corruption in Public Procurement,* Washington, DC: Center for International PrivateEnterprise.

Oboth, M., 2001. *Decentralization and Service Delivery: Constraints and Controversies,* Kampala: Makerere University Library.

Pegg, S., 2015. *The importance of stakeholder engagement in procurement,* International: LinkedIn.

SCI-Network, 2009-2012. *Sustainable Construction and Innovation through Procurement,* EU: Lead Market Initiative for Europe.

SCUBA, 2011-2014. *SCUBA - Self-organising, Cooperative, and robUst\n Building Automation.* [Online] Available at: https://cordis.europa.eu/project/id/288079

Technische Universität Dresden, 2013. *AUTERAS (R) Tutorial.* [Online] Available at: http://www.auteras.de/dokumente/upload/0b6bc_auteras_tutorial_vdi3813.pdf

TechTarget, 2019. *SOAP (Simple Object Access Protocol).* [Online]
Available at: https://searchapparchitecture.techtarget.com/definition/SOAP-Simple-Object-Access-Protocol
[Accessed 2019].
The University of Manchester, 2019. *FaCT++ reasoner.* [Online]
Available at: http://owl.cs.manchester.ac.uk/tools/fact/
The World Bank, 2004. *The OECD / DAC report,* Johannesburg: The World Bank.
TOPAs, 2015-2018. *Tools for cOntinuous building Performance Auditing.* [Online]
Available at: https://cordis.europa.eu/project/id/676760
VDI3805:2011-10, 2011. *Produktdatenaustausch in der technischen Gebäudeausrüstung; Luftfilter.* [Online]
Available at: https://www.vdi.de/richtlinien/details/vdi-3805-blatt-10-produktdatenaustausch-in-der-technischen-gebaeudeausruestung-luftfilter-ueberarbeitung-weiss-auf-weiss-beabsichtigt
VDI3812:2010, 2010. *Assistance functions for the home - determining the need for electrical installations and building-services automation.* [Online]
Available at: https://www.vdi.de/richtlinien/details/vdi-3812-blatt-1-assistenzfunktionen-zum-wohnen-bedarfsermittlung-fuer-elektroinstallation-und-gebaeudeautomation
VDI3813:2011-05, 2011. *Building Automation and Control Systems (BACS) standard.* [Online]
Available at: http://bms-technology.com/pdf/VDI_3813-1.pdf
VDI3814:2017-07, 2017. *Building Automation and Control Systems (BACS) guideline.* [Online]
Available at: https://bauverlag-events.de/wp-content/uploads/sites/11/2017/11/Die-neue-VDI-3814-der-Standard-der-Geb%C3%A4udeautomation-wird-aktualisiert-Prof.-Dr.-Ing.-H%C3%B6ttecke.pdf
W3C, 2019. *Pellet - an open-source Java based OWL 2 reasoner.* [Online]
Available at: https://www.w3.org/2001/sw/wiki/Pellet
W3C, 2021. *XML Technology.* [Online]
Available at: https://www.w3.org/standards/xml/
World_Bank, 2019. *WB: Acqusition of goods and services.* [Online]
Available at: https://data.worldbank.org/indicator/NE.IMP.GNFS.ZS