BIM4EEB

BIM based fast toolkit for
Efficient rEnovation in Buildings

# D6.2 Methods and tools for selecting devices and linking them to the generic model

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

The content of this document reflects only the author’s view only and the Commission is not responsible for any use that may be made of the information it contains.

| | |
|---|---|
| **Programmes** | H2020 |
| **Call for Proposal** | LC-EEB-02-2018 Building information modelling adapted to efficient renovation |
| **Project Title** | |
| **Acronym** | BIM4EEB |
| **Project Grant Agreement** | 820660 |

# D6.2 Methods and tools for selecting devices and linking them to the generic model

| | |
|---|---|
| **Work Package** | WP6 |
| **Lead Partner** | University College Cork (UCC) |
| **Contributing Partner(s)** | TUD |
| **Dissemination Level** | Public |
| **Type** | OTHER: Document, report (excluding the periodic and final reports) |
| **Due date** | 31/03/2020 |
| **Submission Date** | 30/03/2020 |
| **Version** | 1.0 FINAL |

# DOCUMENT HISTORY

| Version | Date | Comments | Main Authors |
|---|---|---|---|
| 0.1 | 19.09.2019 | ToC created for review | Matthias Lehmann (TUD) |
| 0.2 | 24.02.2020 | Chapter 2, 3, 4, 5 added | Matthias Lehmann, Tuan Linh Mai (TUD) |
| 0.3 | 28.02.2020 | Summaries, ch. 1 & 6 added, document cleaning and refinement | Matthias Lehmann, Tuan Linh Mai, Daniel Lunkwitz (TUD) |
| 0.4 | 04.03.2020 | Internal review and corrections | Dr. Andriy Hryshchenko (UCC) |
| 0.5 | 13.03.2020 | Internal review | Jari Shemeikka & Teemu Vesanen (VTT) |
| 0.6 | 19.03.2020 | Minor corrections | Matthias Lehmann (TUD) |
| 0.7 | 20.03.2020 | Internal review and corrections | Dr. Andriy Hryshchenko, Brian O’Regan, Dr. Dominic O’Sullivan (UCC) |
| 0.8 | 24.03.2020 | Minor corrections and explanations | Matthias Lehmann, Daniel Lunkwitz (TUD) |
| 0.9 | 25.03.2020 | Proofreading, corrections | Dr. Dominic O’Sullivan (UCC) |
| 0.10 | 29.03.2020 | Minor corrections and explanations | Matthias Lehmann, Daniel Lunkwitz (TUD) |
| 0.11 | 30.03.2020 | Final proofread, editing | Dr. Andriy Hryshchenko (UCC) |
| 1.0 FINAL | 30.03.2020 | Submission to EU Portal | BIM4EEB Project coordinator (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This deliverable contains original unpublished work except where clearly indicated otherwise.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.

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

# EXECUTIVE SUMMARY

This document describes the research conducted in T6.2 regarding the post-procurement requirement specification for Building Automation and Control systems (BAC). This phase comes after the definition of functional and non-functional requirements of the BAC, for which computer-aided approaches from BIM4EEB were developed as part of T6.1 and reported in D6.1. In this T6.2, new computer-aided solutions and approaches have been developed to enable the automated selection of interoperable BAC-devices to meet specified requirements. These approaches supporting both - the development process of a new BAC as well as the retrofit process of an existing BAC. As the current semantical descriptions for functionalities in BAC are limited to Room Automation functions, the new approaches focus on Room Automation Systems (RAS) but are able to be extended if semantical descriptions for the other parts of BAC are available.

The research from T6.2 was prototyped for validation purposes as part of the AUTERAS software suite for automated RAS design (introduced in D6.1). This suite consists of various supporting tools for different design phases and has been expanded with new supporting functions. This is to assist the user in design tasks during RAS development projects at the post-procurement stage.

As the basis for work conducted in T6.2, an overall workflow was specified. It consisted of engineering tasks necessary for specification of post-procurement requirements. Together with case analysis of different design and retrofit situations, this workflow builds the framework for necessary technical solutions, which can help to reduce the effort for the device selection process.

The research in T6.2 was divided into three main parts to handle major AUTERAS' enhancements to implement the specified workflow.

In addition to supporting the semi-automated definition of the building structure (i.e. interrelation of different locations in a building, e.g. storeys and rooms, not geometry or material information) in accordance with the guideline (VDI3813:2011-05, 2011), a new webservice-based feature has been examined. It supports the automatic import of the initial building structure from the BIMMS (developed in the WP4) and subsequent conversion into a building structure compatible with VDI 3813. This allows full re-use of data previously obtained by other project partners.

In the second part of the research report, the search for interoperable devices to implement the requirements is discussed in detail. This includes an analysis of the current multi-step workflow for selecting appropriate devices and generating a design which describes the “binding of device” functions. This document explains a new device selection approach that aims to allocate all valid designs.  A discussion of the relevance criteria for this approach, as well as integration with AUTERAS, is also included.

The last main part of this report will detail the research and implementation of a new methodology to deal with retrofit scenarios. The design process can also manage the current RAS used and determine if it should be improved or partially replaced during its runtime. This allows realization of backtracking processes along with new system modelling and adaptation of the existing system model.

The AUTERAS suite acts here as the first step of the WP6 tools' chain and it is part of the BIM4EEB's “Tool 4 – HVAC-Design, Operation and Efficiency management”. The capabilities of this tool were tested in the Polish and Italian test sites of the project and where they proved both effective and reliable.

# PUBLISHING SUMMARY

This deliverable D6.2 covers Research and Development (R&D) results and explains the supporting tools for relative stakeholders involved in renovation of existing buildings at the post-procurement phase. The outcome is based on the existing AUTERAS tool suite. That tool suite was initially developed by the TUD and originally designed to outline new Room Automation Systems (RAS) during the planning phase of their development. RAS are part of Building Automation and Control Systems (BAC). The limitation of RAS results is based on the availability of semantical functionality descriptions, which are currently only available for RAS functions.

Furthermore, this report summaries enhancements of the AUTERAS suite in order to fulfil new renovation tasks. Some main improvements include an IFC importing for existing building data – a new methodology to cope with retrofit scenarios and for replacement of the existing device search algorithms. This works for the automatic design generation which had to be revised to enable it to handle the renovating task.

The “IFC importer” enables an AUTERAS’ user to reutilize existing data about those buildings under renovation. It is designed as a “web-service interface” to access IFC documentation (located in the BIM Management System, BIMMS) allowing an automated rebuild of the known building’s structure within the AUTERAS tool. To properly integrate this information with the AUTERAS suite for the new scenario of retrofit, the following had to be considered: new user guidance; handling design generation requests; and representation of results. Thus, the device-search algorithm, as the core component of the automatic design generation process, had to be enhanced in order to handle existing designs on the initial planning phase. The reworked design generation option was integrated into the new workflow to improve users’ experience when selecting, replacing and linking devices in the existing layouts of generic model.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# 1 Introduction

## 1.1. Scope of this report

Procurement is the process of acquiring products or services from a third party. Before this procurement process can start, a stakeholder doing renovation (e.g. automation designer or planner) has to define requirements for a Building Automation and Control system (BAC) to be integrated in a renovated building. After definitions have been completed, those devices to fulfil the requirements must be found and acquired. However, this stage is commonly problematic due to the complexity of the situation. One of the problems is as follows - it is not possible to map one-to-one from a requirement to a device, as most devices can fulfil several requirements. Another problem is tremendous number of different devices with similar abilities on the market.

The planning tool AUTERAS (introduced in the D6.1, being developed by TUD) proposed to solve such problems, partially, before the BIM4EEB project began. Now in the BIM4EEB project, the AUTERAS tool is being further developed by the WP6 and improved with a new device search algorithm to handle the aforementioned complexity.

Originally the "**AUTERAS**" intended to be a tool for "**AUT**omated **E**ngineering of **R**oom **A**utomation **S**ystems", but for handling renovation or retrofit scenarios of Room Automation Systems (RAS) the overall workflow methodology was re-examined in the WP6. The limitation on RAS as part of BAC (see Figure 1 and also sec. 1.1 of D6.1) highlights the current limitation of standardized semantical description methods for automation functionalities which currently only exist for RAS. If such description methods are also available for the other parts of BAC, the AUTERAS suite is also able to address these parts.

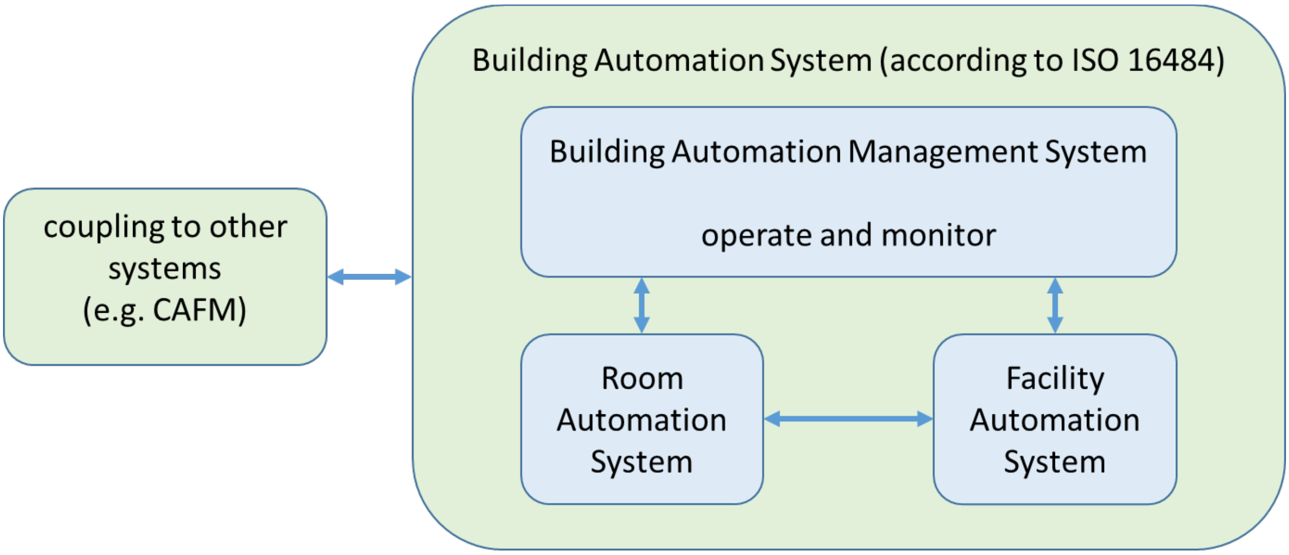

Building Automation System (according to ISO 16484)
Building Automation Management System
operate and monitor
coupling to other systems (e.g. CAFM)
Room Automation System
Facility Automation System

**Figure 1: Relation of the different parts of BAC according to ISO 16484**

With the extensions of BIM4EEB, the AUTERAS suite can now also extend (or change) existing RAS design for buildings under renovation.

The AUTERAS suite needs an input of the building' structure to perform, but initially there was only manual input integration. In this report the new importing mechanism will be described, which takes this input information from the BIMMS (developed in the WP4) via RESTful interface. Thus, users don't need to input the building' structure manually. The scope of this report is therefore including detailed explanation of these improvements and summarising the AUTERAS toolkit development definition.

## 1.2. Relevance to previous deliverables

This report D6.2 is the direct successor of the D6.1 “Open format and formalised requirements specification for procurement”. The results from that deliverable were directly used and further processed.

Furthermore, this D6.2 report is predecessor of the D6.3 “Tool for constraint checking BAC topologies VS building codes”. In practical terms, after the completion of BAC functional design, a constraint checking process could be started with other tools developed by the WP6.

Nevertheless, there are strong connections to other work packages of the BIM4EEB project, e.g. the WP4 “Development of BIM management system”. Specific information will be taken directly from the “central data store” aka BIMMS to reduce task repetitiveness and data inconsistencies within different tools of the BIM4EEB WP6.

Moreover, the underlying 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” and D3.2 “A refined, integrated domain ontology for occupants, building services, building energy and acoustics” were considered to ensure good integration of the T6.2 results into the complete structure of the BIM4EEB project.

# 2 Workflow from early requirement definition to selecting devices

## 2.1 General workflow of planning BAC systems in renovation scenarios

The general lifecycle of buildings and especially Building Automation and Control Systems was introduced in the D2.1 referencing the EN16310 standard. In most lifecycle models, only the forward direction from the initiative up to the demolition (end of life) is considered (i.e. the lower arrows in Figure 2). However, in practice there are many backward steps necessary. They are most easily envisaged at the design phases, for example in situations where mistakes in earlier design phases are identified. In that case re-planning is needed for planning processes to be continued. It is important to say that several steps backwards are possible if necessary (i.e. upper arrow block in Figure 2).

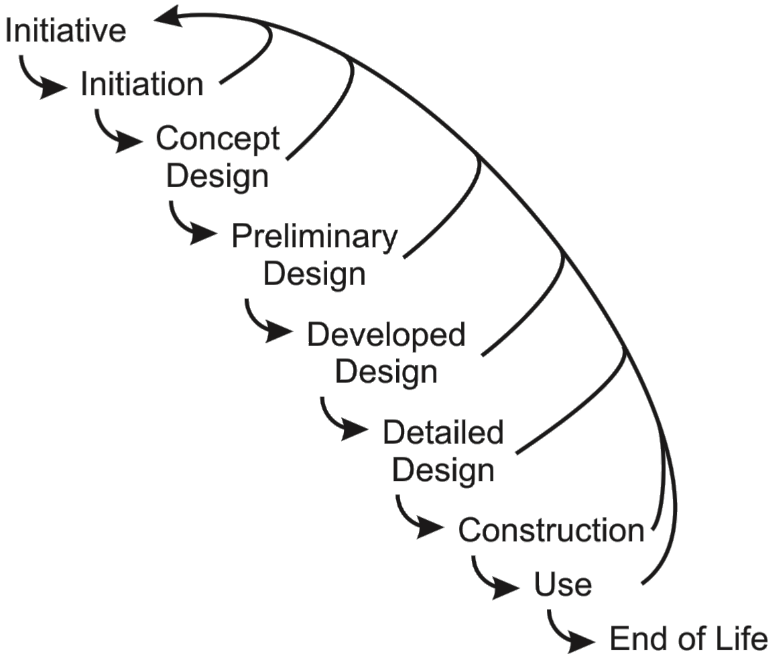


**Figure 2: General Lifecycle of Buildings**

Another important stage is renovation or retrofitting of existing and working systems (indicated by the phase “Use” in Figure 2). The main reason for those scenarios is the long working life of buildings and their systems. In (Kleiber, Simon, & Weyers, 1998) we can see approximate lifetimes for different building types, see in Table 1 below:

**Table 1: Working lifetime of several building types**

| Building type | Working lifetime [years] |
|---|---|
| detached house (according to the quality) | 60-100 |
| prefabricated house in solid construction | 60-80 |
| Prefabricated house in half-timbered and panel construction | 60-70 |
| settlement house | 50-60 |
| simple house (solid) | 50-60 |
| apartment building (privately financed) | 60-80 |
| apartment buildings (social housing) | 50-70 |
| mixed-use houses with a commercial rental income share of up to 80% | 50-70 |
| administrative and office buildings, schools, kindergartens | 50-80 |

| | |
|---|---|
| commercial and industrial buildings with flexible and future-oriented execution | 40-60 |
| gas stations | 10-20 |
| shopping center / hypermarkets | 30-50 |
| hotels / sanatoria / clinics | 40-60 |

The building envelope itself, usually, has a long working lifetime, but internally installed automation systems tend to age much faster. In the literature (ASHRAE, 2019), an average lifespan of 15-22 years is mostly stated. This means that a building and it's integrated systems will be renovated several times. This renovation can be required for several reasons and can result in different strategies:

| **Reason for renovation** | **Strategies** |
|---|---|
| replacement of failed devices | working devices will be not replaced |
| enrichment of quality, e.g. comfort, energy | old system will be unchanged, only additional installations |
| | old system will be changed and additional installations |
| | old system will be removed and completely new installations |

All these possiblities have a common property: new parts must be planned (in case of partial renovation the new parts should be harmonized with the old system parts; in case of complete renovation it is similar to construction of a new automation system).

The tool suite AUTERAS for planning and designing Room Automation Systems was enhanced with new methodologies for coping the renovation aspect in the BIM4EEB WP6. The underlying algorithms were improved to handle larger systems (with increasing complexity) and to consider non-functional requirements (described in the D6.1 report) for RAS.

Therefore, in the next section of this report the workflow of the AUTERASsuite will be shown. Moreover, in the following chapters the enhancement of AUTERASdesign device search algorithms will be explained in detail.

## 2.2 Workflow of AUTERAS

As it was previously stated in the D6.1, AUTERAS consists of five parts, i.e.:

- **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.

The AUTERAScatalog and AUTERASstore are background systems without any GUI. The communication with these tools is web-service based. The AUTERASplan' and AUTERASdesign' functionality is hosted on a server of TUD. The GUIs of both these tools are encapsulated in a single GUI of the AUTERASsuite. The workflow and all descriptions below use these GUI for showing the enhanced functionalities of the AUTERAS developed in WP6 of the BIM4EEB. As the workflow of AUTERASsuite is very complex and the results might be very hard to understand for non-BAC-experts, TUD has also established a web-GUI special for end-users. This GUI (available under www.auteras.de) is named as the "AUTERASlite".

The AUTERASlite uses (on the background) the functionalities of the AUTERASplan and AUTERASdesign hosted on TUD's servers. If only the name "AUTERAS" is used in this report - all parts of the AUTERAS are meant.

The general workflow of the AUTERASsuite is depicted in Figure 3.

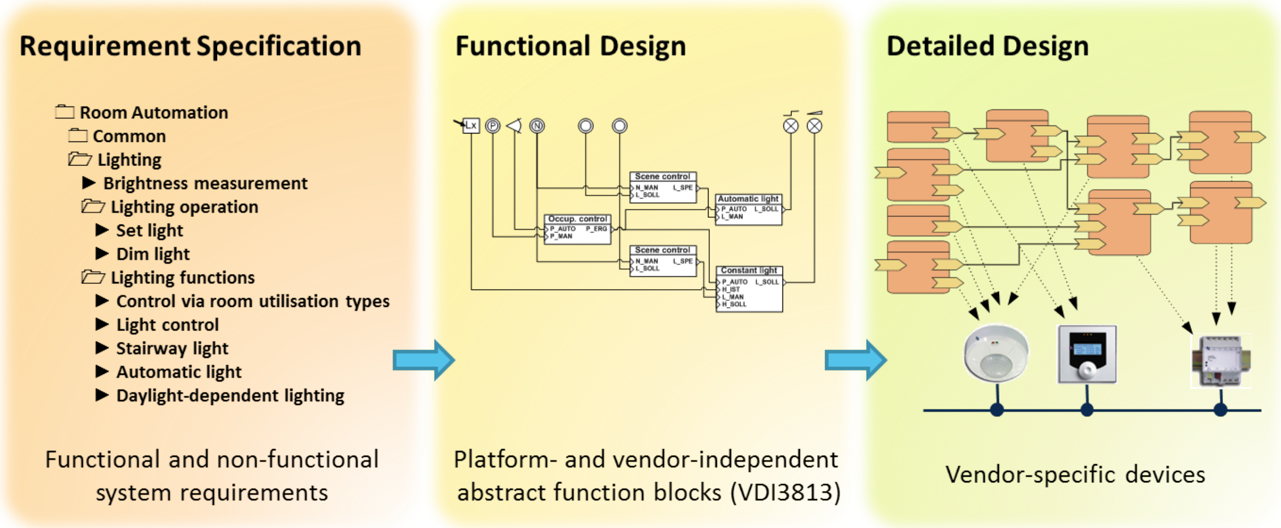


**Figure 3: General workflow of the AUTERAS**

First step of working with the AUTERAS is to survey basic building structure with the AUTERASplan. This is needed to assign the later resulted room templates to the different locations of the building. After this the process of designing RAS can start with a survey of the user requirements. The functional requirements will be transformed into the unambiguous room automation scheme (RAS-schematics) based on the function set defined by the German guideline (VDI3813:2011-05, 2011). Such a RAS-schematic are vendor- and platform-independent so they can be used for procurement processes. After that, specific devices that fulfil the user requirements will be searched with AUTERASdesign. The AUTERASdesign uses the AUTERAScatalog for this process – a background repository with semantical device descriptions.

Assuming that in many large buildings various rooms are equipped the same, the AUTERASplan works with a template-based approach. In office buildings there are mostly offices, sanitary rooms, server rooms and meeting rooms and so on. In apartment buildings there are rooms like kitchen, bathroom, living room, floor etc. To simplify design process the user needs only to describe the room types and attach them to the different rooms. In the AUTERASplan' workflow, the user has to create the requirement templates and to attach them to the different rooms. When the RAS-schematics or detailed designs are generated - they will be automatically assigned to the corresponding rooms.

The original goal of the AUTERAS was to assist users in their planning processes of Room Automation Systems in new buildings. However, those processes can also be used for retrofitting scenarios. There are three different scenarios that can be assumed:

- In the old building there is no RAS installed → the renovated building gets a new RAS → the original workflow of the AUTERAS can be used;
- In the old building there is a RAS installed, but should be replaced completely → the renovated building gets a new RAS → the original workflow of AUTERAS can be used;
- In the old building there is a RAS installed, but should be partially re-used → new strategy in the AUTERAS to cope with this case is needed → described in chapter 5 of this report.

# 3 Building structure in the AUTERAS and import from IFC

In this chapter, the structure of a building used in the AUTERASsuite is explained in comparison with well-known building structure models defined in VDI 3813 or IFC standards.

In context of the AUTERAS application, the "building structure" term means "hierarchy of different locations in a planning project". All other information about structures (e.g. wall materials or geometry data) is not relevant to the AUTERAS and not included into resulting structure model.

In the second part of this chapter the new importer of building structures from IFC files via the REST-service (developed by the WP4 of BIM4EEB) into AUTERAS is presented.

## 3.1 Building structure of the AUTERASsuite

The room automation system-planning tool-suite AUTERAS needs basic building structure as an input for design process to begin. This need is occurring from the template-based workflow of the AUTERASsuite described above. In the standard (VDI3813:2011-05, 2011) a basic shell model for building structures is given, see Figure 4 below.

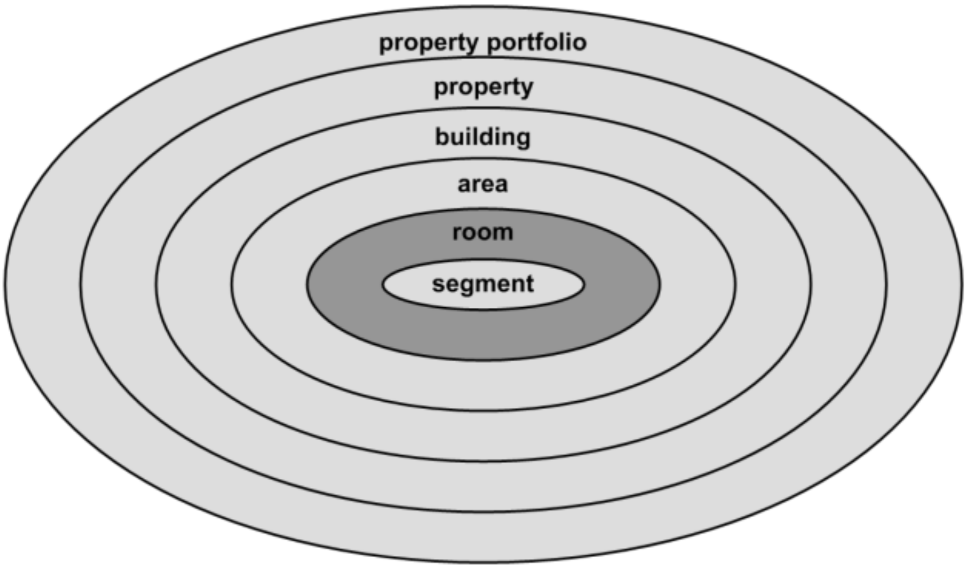


**Figure 4: Building structure model in VDI 3813**

The AUTERAS uses quite a different model, albeit inspired by the VDI 3813-model. The reason for the differences is based on better understanding of a building' structure by the user during renovation processes. Moreover, there are some room types and other aspects that were not included in the VDI 3813, e.g. outer structures like facades. The resulting structural model of AUTERAS is displayed in following Figure 5 on the left. In the middle of the Figure 5 there are corresponding elements of the VDI 3813 displayed. On the right-hand side there are IFC elements correspondingly listed. The same issue exists with IFC as with VDI 3813, that is some building elements could be modelled only using the generic "IfcBuildingElement" type, which can be anything.

| AUTERAS building structure model | corresponding VDI 3813 - elements | corresponding IFC - elements |
|---|---|---|
| project | property portfolio | IfcProject |
| site | property | IfcSite |
| building | building | IfcBuilding[Element] |
| section | | IfcBuilding[Partial] |
| facade | | IfcBuildingElement |
| storey | | IfcBuildingStorey |
| area | area | IfcSpace[Complex] |
| room | room | IfcSpace[Element] |
| segment | segment | IfcSpace[Partial] |

**Figure 5: AUTERAS building structure model**

Thereby a project represents a container of all working data in AUTERAS for a single self-contained planning order, which can be stored to a file or into the system information model (introduced in D6.1). A project contains sites, e.g. different campuses of a university. A site is a container for buildings descriptions. A building consists of different sections – the building wings. The sections contain facades at the outside of the building and storeys inside. Assuming sections are only of interest in the case of a very large building, it is also possible to omit them and to put facades and storeys directly into a building. An area as the sub-element of a storey groups different rooms in a storey, e.g. between two fire doors. The innermost elements are rooms inside areas. Within the planning process of larger buildings, it is often unclear how many rooms will be necessary during later operation, so in those areas there are different segments that will be defined. A segment, for example, could consist of a window, a heater, a light system and could be 3m wide. Now this area will be filled up with such segments. Afterwards, if the room partitioning is clarified, one or more segments can be combined to rooms and separated from the others with lightweight construction walls.

In the AUTERASsuite there is a single view implemented where the user can see the existing building structure and can edit it manually by right-clicking on an element, remove it or define a new sub-element.

## 3.2 Import of IFC-files into AUTERASsuite

To re-use existing models built with other tools from BIM4EEB work packages, the new importing functionality was introduced in the AUTERASsuite. The existing building structures are available in the BIMMS, e.g. those developed by the WP4 activities. The BIMMS provides the RESTful-service interface allowing to retrieve the building structures from it.

To create a new project, the AUTERASsuite has an option of "New Project from BIMMS" creation (see Figure 6 depicting this). Afterwards the AUTERASsuite calls automatically the REST-interface of the BIMMS with some user information:

- getUserToken(username, password) – gets a token for later use, the username and password are predefined and fixed;
- getUserProjects(tokenUser) – gets information for all projects in BIMMS.

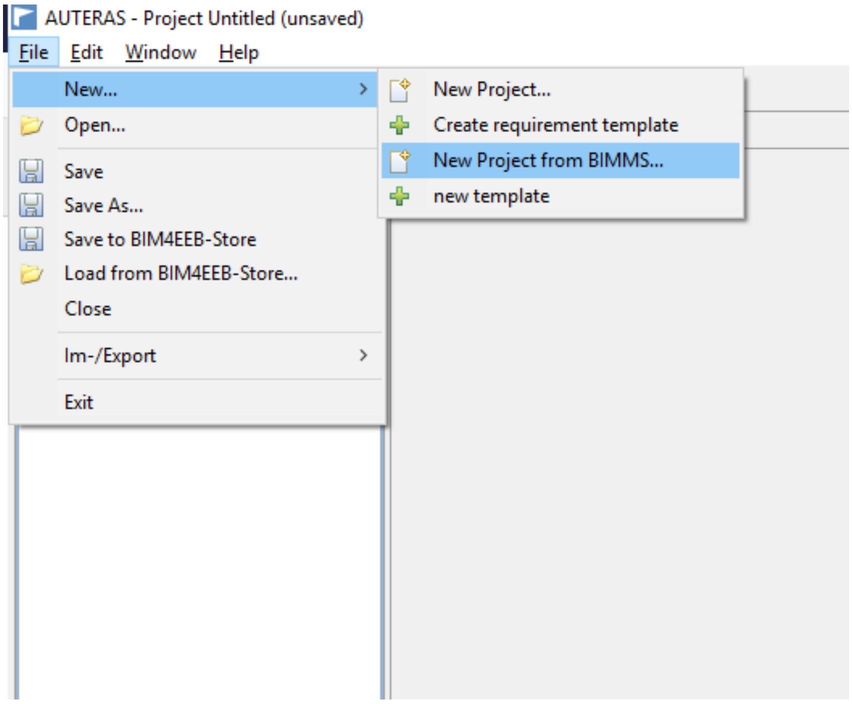

AUTERAS - Project Untitled (unsaved)
File Edit Window Help
New...
Open...
Save
Save As...
Save to BIM4EEB-Store
Load from BIM4EEB-Store...
Close
Im-/Export
Exit
New Project...
Create requirement template
New Project from BIMMS...
new template

**Figure 6: AUTERAS' menu - New Project from BIMMS**

Now, a new dialog (see Figure 7) will be opened for the user to select the information for the project to load from BIMMS. In the first ComboBox all the project names (determined from BIMMS before) will be listed. The user can select one and in the background those available IFC-files in BIMMS are determined via the BIMMS-REST-method getProjectIFC(tokenUser, project). The user can select one and click OK. After this AUTERAS calls the last method - getIFCHierarchy(tokenUser, project, fileName). This function returns a JSON-list with all locations in the selected IFC-file and the AUTERAS recreates the building hierarchy (see Figure 8) with those correspondences displayed in Figure 5 in a new AUTERAS project.

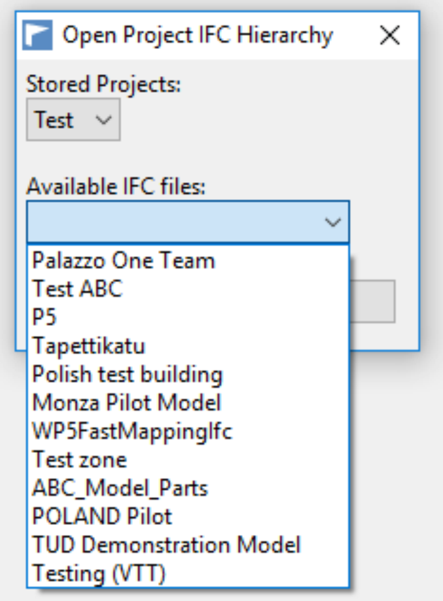
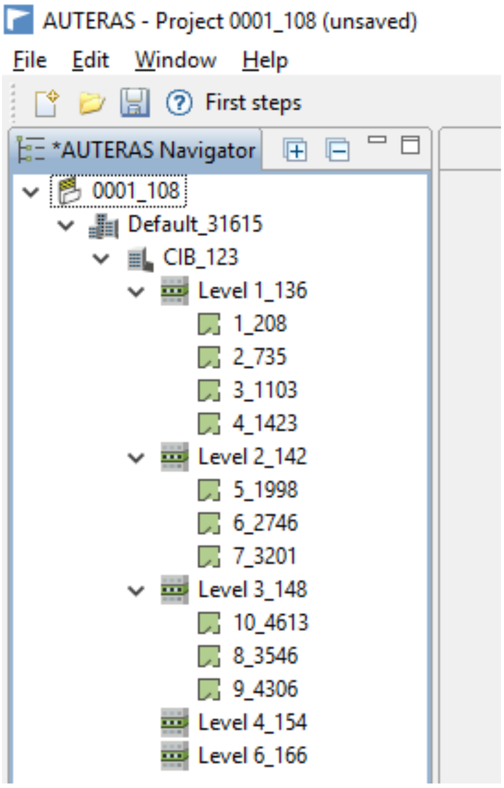


**Figure 7: Dialog for selecting project information**



- 0001_108
  - Default_31615
    - CIB_123
      - Level 1_136
        - 1_208
        - 2_735
        - 3_1103
        - 4_1423
      - Level 2_142
        - 5_1998
        - 6_2746
        - 7_3201
      - Level 3_148
        - 10_4613
        - 8_3546
        - 9_4306
      - Level 4_154
      - Level 6_166

**Figure 8: Building structure loaded from BIMMS**

The user is now ready for fast planning and designing RAS with the AUTERASsuite. After this, the user can store results (see AUTERASsuite's menu bar in Figure 6) for further use by other tools in the system information model (based on AUTERASstore) described in the D6.1 report.

# 4 Device search functionality

The goal of the device search (performed by the AUTERASdesign tool) is to determine real devices, which are available on the market (stored in the AUTERAScatalog), to fulfil the user-defined functional and non-functional requirements of the RAS in design / renovation. To find such solutions there are many issues to consider, i.e.:

- all requirements must be fulfilled;
- the devices must be interoperable to ensure their combined functionality:
  - in cases they use the same communication protocol (for RAS mostly fieldbuses like LON or KNX) it must be checked if the data point types of those devices need to be bound together;
  - in cases where devices should be bound together, but operate using different communication protocols, then suitable gateway must be integrated between them;
- the resultant solutions should be ranked by user-predefined criteria, e.g. suitability, non-functional criteria like price, energy consumption etc.

To solve these issues, the design process results in a combinatorial optimization problem with an exponential complexity. In (Lehmann, Schäfer, Sattler, & Kabitzsch, 2018) an estimation of the search-space size is described. In worst case the amount of possible solutions $sol$ is

$$sol = devs^{funcs}$$

with the number of available devices $devs$ and the amount of functions $funcs$ in the underlying RAS-schematics.

In practice, an average RAS-scheme with HVAC, lighting and shading functions includes around 20-30 functions for a single room. In addition, there are hundreds of room automation devices on the market to fulfil these functions. This results in a very large search space; and a compromise between calculation time and quality of the solutions must be found.

The original device search process of the AUTERASsuite is based on the evolutionary algorithm' use. The main advantages of this type of algorithms are:

- they work on a set of individuals (population-based algorithms), so the result is not only one solution, but a set of solutions;
- the individual set can contain invalid solutions (which don't fulfil all user requirements);
- the algorithms find every solution (not the global optimum or invalid solutions only);
- the algorithms will approach to the optimal solution (similar to gradient method);
- the algorithms are able to break out from local optima with random mechanisms;

The general workflow of such evolutionary algorithms is depicted in Figure 9.

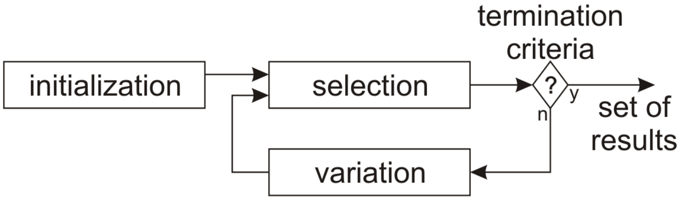


**Figure 9: Workflow of evolutionary algorithms**

It starts with an initialization phase to generate the “start items” population. Mostly these start items will be generated randomly. An enhancement will be done with the use of pre-optimized start items generated, for example, with the “Greedy-algorithm” (Ploennigs, Lehmann, Wollschlaeger, Mai, & Kabitzsch, 2014).

After that, the actual evolutionary process starts. It is inspired by evolution in biology.

Now, the items in the population set will be ranked by their suitability (if all functional and non-functional requirements fulfilled). The items with poor suitability will be sorted out – they have to “die”. So the population set will be getting smaller.

In the next step, it will be checked if the remaining solutions are suitable enough for user’s requirements. If this is the case - the algorithm can be stopped. Another termination criterion is the number of lifecycles. In cases in which the algorithm does not reach the required suitability value - it can happen that the algorithm will work infinitely. To avoid such situations, an upper bound of lifecycles should be set (in deviation from the analogue biological processes).

In the variation step, new items will be created from the existing items – they will be “born”. The variation step is divided in two different parts – the recombination and the mutation.

- In the recombination part, new items will be generated from the existing items by taking good properties from them and setting up new items (gradient-like optimization);
- In the mutation part some items will be generated randomly. This is to break out from local optima and allow searching in other areas of the search space.

After the variation step, the number of items in the population set should be the same as before the selection step (also in difference to the biological evolution).

After that, the next selection step can run, and the loop is closed. In contrast to the simple general workflow (see Figure 9) and stated advantages, the algorithm can also have some difficulties, namely:

- The pre-set number of iterations (lifecycles) could be too small, so the algorithm cannot reach the global optimum;
- Finding a suitable balance between the recombination and the mutation. If the mutation part is too large - the whole process becomes random; if is it too small - the process is not able to break out from local optima;
- Ensuring sufficient diversity of the items. The recombination can tend to generate only clones if items are too similar (but the mutation can work against it);
- Appropriate choice of suitable strategies for items’ selection on the variation step (there are many common strategies available (Deb, 2001) as well as some strategies particular to the design process of room automation systems (Özlük, 2013);
- Finding definitely appropriate solution – valid solutions lie very wide apart from each other in the search space. Even small changes of a valid item (in the variation steps) results in a high probability of an invalid solution;
- The algorithm scales insufficiently with the increasing complexity of the systems to design. In large

commercial buildings a lot of devices should be integrated (up to approx. 10,000 devices), but many of them are the same (e.g. same sensor in each room). In contrast, for smaller residential buildings the complexity of the design process is increasing due to a large diversity of planned devices and systems to be installed (with different communication protocols, different communication media etc.).

To circumvent these difficulties and to reduce efforts required for the planning process, a new approach to devices' search was investigated and rudimentary implemented for a general validation in the earlier project (TOPAs, 2015). At that time, it became apparent to follow this new approach which covers the issues associated with high complexity without the difficulties of the existing evolutionary algorithm.

In the BIM4EEB WP6 the new approach was integrated completely with the AUTERASsuite. The old evolutionary algorithm is now replaced with the fallback algorithm described in the following section of this report and originally created in TOPAs. In addition, the new approach was extended in its functionality to cope with a practically relevant number of devices and RAS-schematics.

The new approach is not following "the normal" workflow when searching in the search-space for valid solutions. The new approach tries to find invalid solutions and remove them from the search space. At the end, when all invalid solutions are removed, the resulting solution set contains only valid solutions. The main advantage of this procedure is that all valid solutions can be found. The new algorithm tries to reduce the search space at each step as much as possible to manage its complexity.

The new algorithm consists of four steps displayed in Figure 10. They will be described more in detail in the following section of this report.

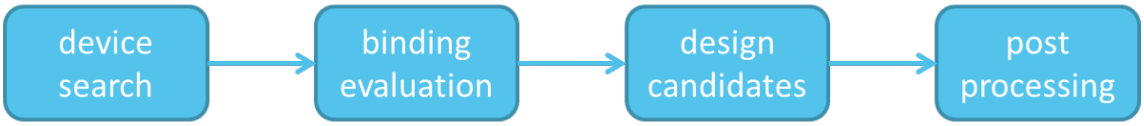


**Figure 10: Workflow of the new approach for device' search**

As an example for the following description, the typical lighting RAS-schematics are used (see Figure 11). At the upper-left corner there are six different sensors' functions; on the upper-right corner there are two actuators' functions integrated. The larger boxes in the middle are displaying the control functions required.

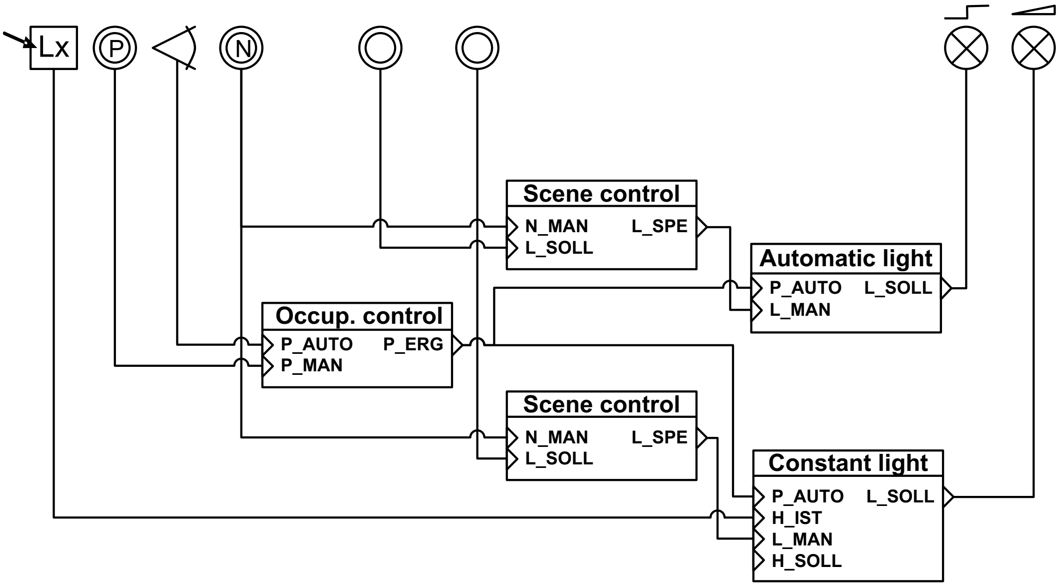


**Figure 11: Example of the typical lighting scenario**

## 4.1 New device searching approach

In this section, the four steps of the new selection approach will be explained in more detail.

### 4.1.1 Device Search

In the first step of the new algorithm for each function of the underlying RAS-schematics, suitable devices will be searched for from an existing semantical device description repository (i.e. AUTERAScatalog). The result of this step are sets of devices for each function. These sets of devices can and should be reduced furthermore by removing all those devices which cannot fulfil the component-related requirements (requirements that depends on only a single function of the schematics, e.g. the colour of the device, or it should be wall-mounted etc.). The more of such criteria the user pre-defines, the smaller the resultant sets of those devices that will appear. No design-related requirements (e.g. "all devices of the design solution should have the same colour") can be checked at this point.

The following Figure 12 illustrates this step for a single function. The resulting set for a function is much smaller than the original set of available devices in the repository.

If a set of devices for at least one function in the RAS-schematics is empty - it is clear that no valid solution is possible to find given that not all functions of the RAS-schematics can be realized. In this case the device search can be completely stopped. The user can now have a look at his/her defined requirements, to adjust them and try it again.

The next step in the algorithm needs only to work with minimized sets of functions and not on the large number of devices.

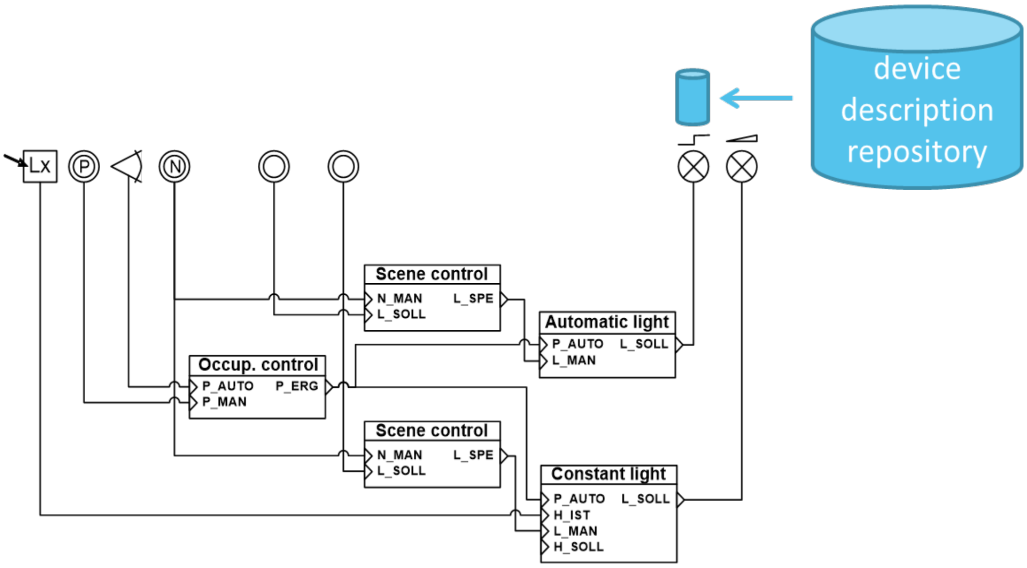


**Figure 12: First step - Device Search**

### 4.1.2 Binding evaluation

In the second step of the new approach, the bindings in the RAS-schematics will be considered. A binding connects two functions. For the two functions, only the significantly reduced sets of devices from step 1 must be considered. This is a pairwise comparison between each device for the source function of the considered binding and each device for the destination function. If they are interoperable - this directed pair of devices will be stored in a solution set for the considered binding. At the end of this step, there exist sets of device pairs for each binding.

A valid solution must have at least one pair of devices for each binding in the underlying RAS-schematics. If not, the required functionality displayed in the RAS-schematics can't be fulfilled.

The following Figure 13 illustrates this process for a single binding. The light blue cylinders at the functions should display the results of step 1. The green and purple cylinders are the subsets of them which can be paired. Nevertheless, the result (large light blue cylinder) is not the pair of these both subsets but the set of pairs of two elements (one of the green and one of the purple cylinders) which can be connected together.

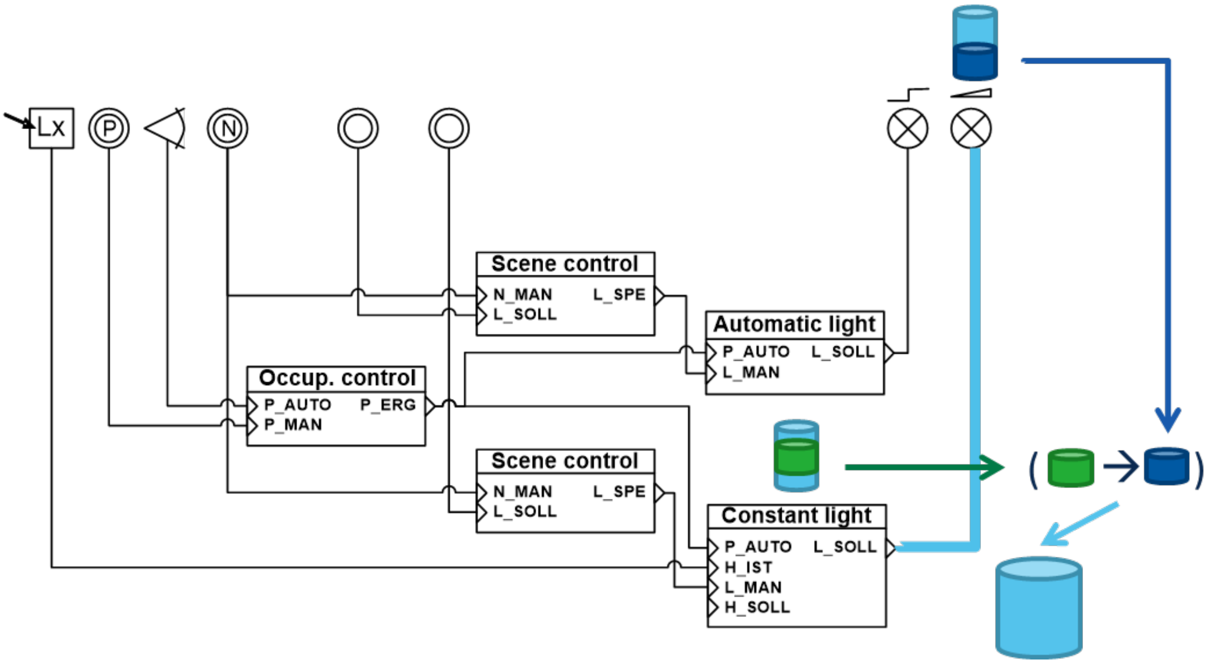


**Figure 13: Second step - Binding Evaluation**

This step reduces the search space by removing all device combinations that are not interoperable according to the binding. The sets can be reduced further by considering design-related criteria. If a criterion is "all devices of the design solution should have the same colour" - all pairs of devices with different colours can also be removed.

Similar as in the first step, if there is an empty set of device pairs for at least one binding - the whole device search process can be stopped, because no valid solution can be found for the processing of the RAS-schematics.

After this step, the results from step 1 can be discarded as they are not needed anymore.

### 4.1.3 Assembling Design

In the third step of the device search process, the complete designs will be assembled. The device pairs for single bindings (result of step 2) have to be combined to build up the whole graph of a solution. Two neighbouring bindings fulfil at least one common abstract function. So it has to be checked if in the device pair sets of the neighbouring abstract bindings a combination can be found which uses the same device for the common abstract function.

Therefore, a randomly chosen binding from the RAS-schematics will be taken as a pivot element. All realizations for the pivot binding will be put into the solution set of this step. These are the starting solution candidates for the upcoming steps. Thus, the next binding for consideration must be chosen to enrich the solution candidates in the solution set until they fulfil the complete RAS-schematics. There are four scenarios possible for selecting neighbour binding of the pivot binding.

1. Neighbour’s destination is the same as pivot’s source, see Figure 14 below:

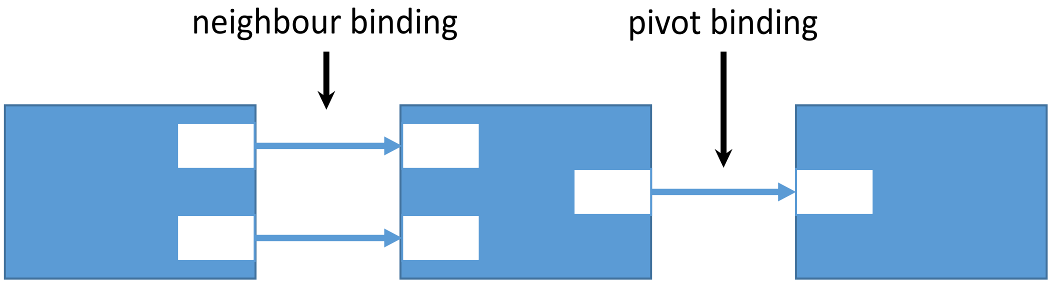


**Figure 14: Neighbour’s destination is same as pivot’s source**

2. Neighbour’s source is the same as pivot’s source, see Figure 15 below:

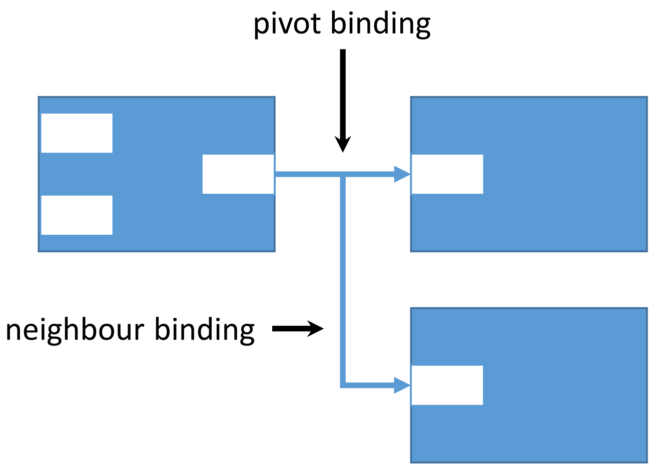


**Figure 15: Neighbour's source is same as pivot's source**

3. Neighbour’s source is the same as pivot’s destination, see Figure 16 below:

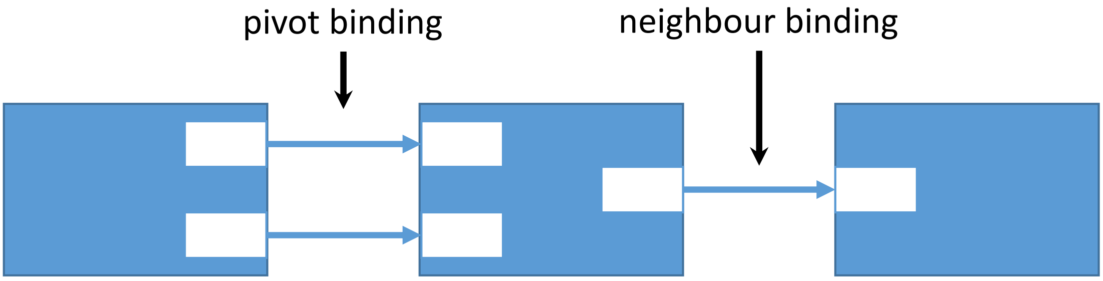


**Figure 16: Neighbour’s source is same as pivot’s destination**

4. Neighbour’s destination is the same as pivot’s destination, see Figure 17 below:

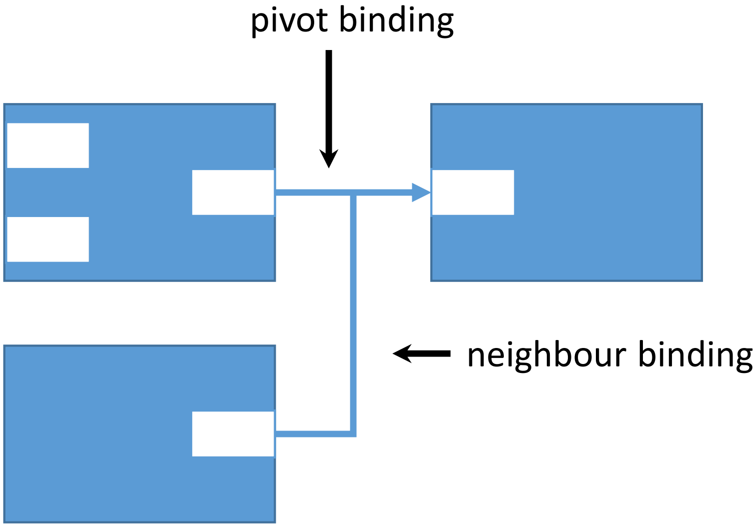


**Figure 17: Neighbour’s destination is same as pivot’s destination**

All pairs of devices which should be connected to realize an abstract connection (result of step 2) from the neighbour binding must be compared with each solution candidate in the solution set.

In case 1 (see Figure 14), the source function of the pivot binding in the solution candidate must be the same as the destination function from the neighbour binding’s realization. If this is the case - the solution candidate can be enriched with the neighbour binding’s realization. If not - this neighbour’s realization can be ignored.

For case 2 (see Figure 15), the source device of the neighbour binding and the pivot binding must be the same.

Case 3 (see Figure 16) is similar to case 1, but the pivot’s destination device must be the same as the source device of the neighbour binding’s realization.

Last case 4 (see Figure 17) is similar to case 2, but the destination devices must be the same.

If there is no neighbour binding corresponding to a solution candidate that can be found - this solution candidate can be removed from the solution set, because there is no possibility to enrich it to a full solution.

In Figure 18 this mechanism is displayed for a single step of two neighbouring abstract bindings with the cylinder representation like in step 2.

After considering all neighbour’s realization possibilities, the neighbour binding becomes the new pivot binding and the enhancement of the solution candidates starts again until the full schematics-graph was realized with suitable, interoperable devices.

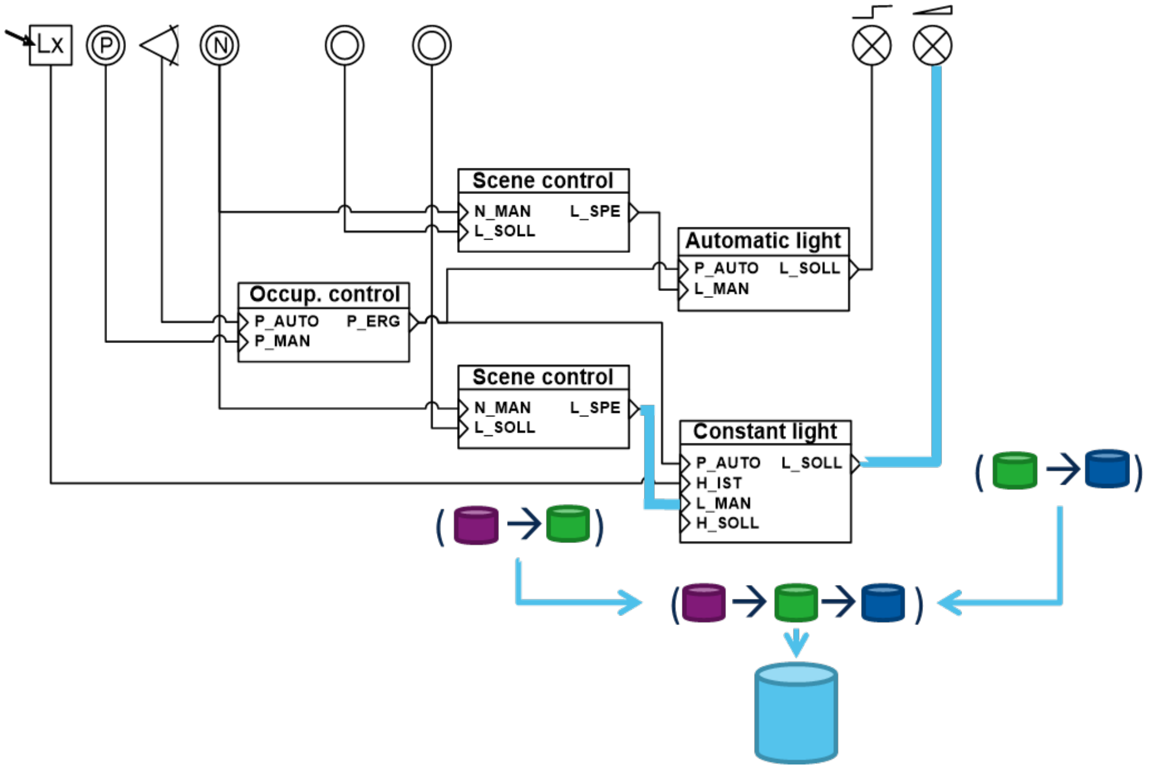


**Figure 18: Third step - Assembling Designs**

In each step of the solution candidate enhancement, the predefined design-related criteria can be checked. If a solution candidate does not fulfil every design-related criterion - this solution candidate should be removed from the solution set immediately.

Also, in each step of the enhancement it must be checked if the current solution candidate has devices that are also able to realize the neighbour binding too. In this case they can be re-used to minimize the number of devices in the solution candidate.

At the end of this process, the solution set consists of all valid solutions. If the solution set is empty, no valid solution is possible. The results of step 2 are not needed anymore and can be discarded.

### 4.1.4 Post-processing stage

At this stage it can be guaranteed that the result set contains only complete and valid solutions. These solutions can now be sorted to present the best solutions first to the user. For this sorting process, there are different options possible to be integrated, i.e.:

- Pareto-Sorting;
- Weighted-Sum-Sorting.

In the AUTERAS it is the Weighted-Sum-Sorting option that is integrated. The background of this decision is that this option is more intuitive for most users. In addition, they are able to change the weights for their requirements more easily.

## 4.2 Relevance of criteria

As it was previously presented, the user can reduce the search space drastically by defining additional requirements (to the point where no solution remains). Conversely if the user defines only a few requirements, how is the complexity handled in such cases?

For the scenario of only a few defined criteria, the search space can also be reduced by definition of virtual devices, which represent similar devices.

In the first step of the AUTERASdesign algorithm (step 1 – device search) all devices of a solution set for the functional RAS-schematics can be merged into one virtual component, which are structurally similar. Structural similarity means that they have the same functionality and the same data points to the outside. In addition, they must have the same properties with respect to user-predefined requirements. For example, if the requirement is “all devices must have the same colour”, then devices can be merged to a virtual component only if they have the same colour.

This approach of merging devices to virtual devices can drastically reduce the search space. In practice, there are many similar devices. For example, a manufacturer provides the “same” device in different colours or forms. This merging devices process reduces the search space which is important in the case where the user defines only a few requirements.

The second and third step of the algorithm can now work on the reduced sets of devices. After the third step the virtual components must be divided into their underlying (formerly merged) real devices - to get the real solution set back.

As a result, a reduction of the search space is possible in cases of both many user-defined requirements, as well as in cases of only a few user-defined requirements.

## 4.3 Integration in AUTERAS

Based on the previous sections of this report, it is possible to suggest that user-defined requirements have the power to result in no valid solutions being generated. For that case the old evolutionary algorithm is still integrated in the AUTERAS operational logic. It is able to find “good”, but invalid solutions. The user can analyse them and decide if some requirements can be designated as not as important, even though they are blocking valid solutions, and thus remove those requirements.

The following Figure 19 shows the mechanism of how the new and old algorithms are both integrated into the AUTERAS.

Another possibility is, especially for powerfull multi-core computers, to start the evolutionary algorithm at the same time with the new approach. In cases where the new approach finds solution(s) – the evolutionary algorithm can be stopped. In the other case of no valid solution is possible - the evolutionary algorithm has already approximated with some “good” but invalid solutions.

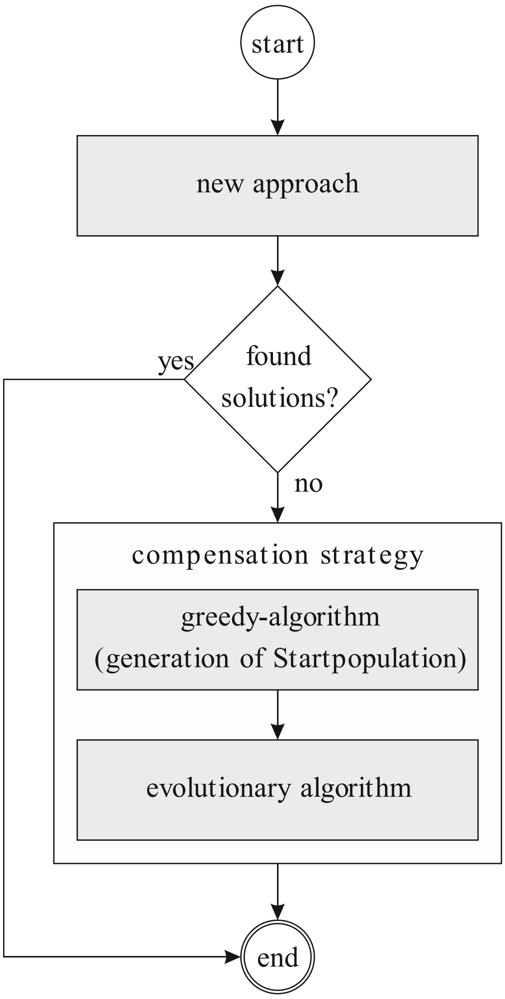


**Figure 19: Integration of device search algorithms in the AUTERASsuite**

## 4.4 Summary to Chapter 4

The new approach for the device search is able to find all valid solutions. The main advantage over the old evolutionary algorithm is that in the case of new approach, it determines there whether there are valid solutions possible. By using the evolutionary algorithm only, this is not the case. If there are no valid solutions in the population set of the evolutionary algorithm - this does not necessarily mean that no valid solutions are possible. This means that the algorithm has no valid solution found until this time/step.

The new approach is mainly constructed concerning fast minimization of the search space to handle the exponential complexity of the device search problem. It is a very memory-consuming process in case where many solutions can be found. There are two different strategies that were explained to reduce this problem of exponential complexity.

The old solution, based on an evolutionary algorithm, was not removed from AUTERAS to handle scenarios where no valid solutions are possible.

# 5 Retrofitting and extension of existing systems

## 5.1 Mapping between model layers as the basis for computer-aided extension of existing automation systems

The design process of RAS aims to create functional automation systems that fulfil different functional and technical user requirements. The automation system is created step by step through different abstraction levels (aka. model layers). The information on each layer was generated based on the information of the other more abstract layers created before. For this purpose, there is no need to store the information about mapping between corresponding objects on different model layers.

To support retrofitting and extension of the existing systems, it is necessary to include this type of information into the model. The mapping between the model layers is illustrated in Figure 20.

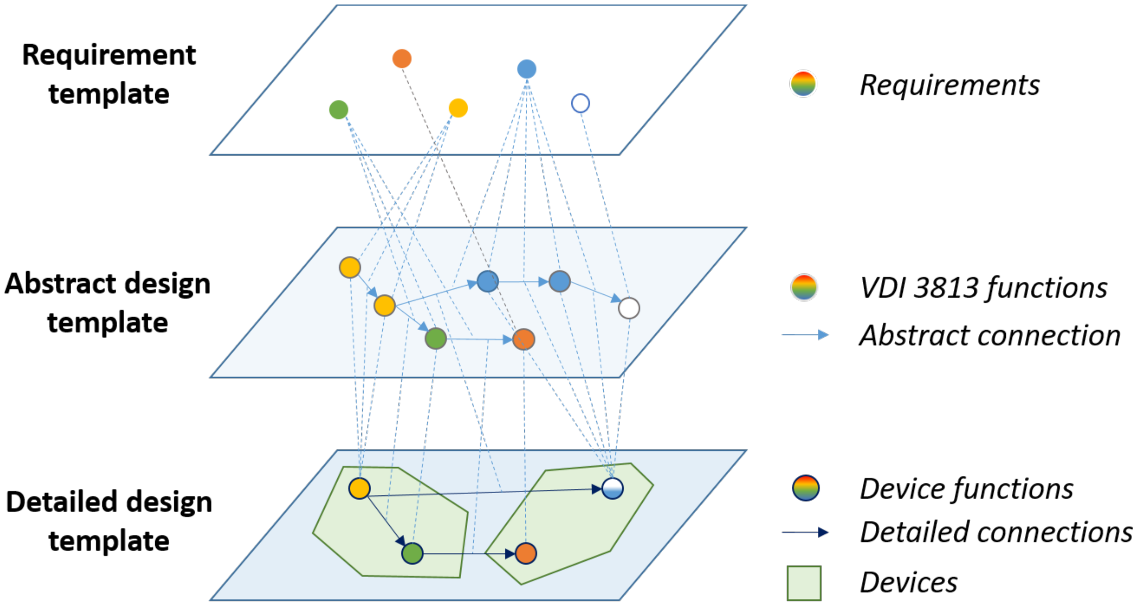


**Figure 20: Mapping between model layers**

Requirements in the requirement template will be linked to components in the abstract design template that directly realize the corresponding requirements. These components are abstract function blocks and abstract connections between the function blocks that are required to realize the corresponding requirements.

Mapping between the abstract design template and detailed design template is done firstly by mapping between functions of devices in the detailed design template and the VDI 3813 functions in the abstract design template that are implemented by the abovementioned device functions. The abstract connections in the abstract design template are realized by detailed connections for data transfer between device functions or by internal connections within each device function. In case of function-external connections, the detailed connections can be device-internal or device-external.

This mapping between the model layers is bidirectional and created during the RAS design process.

This new source of information allows different search functions within the current system model, which is necessary to analyse for system extensions, for instance:

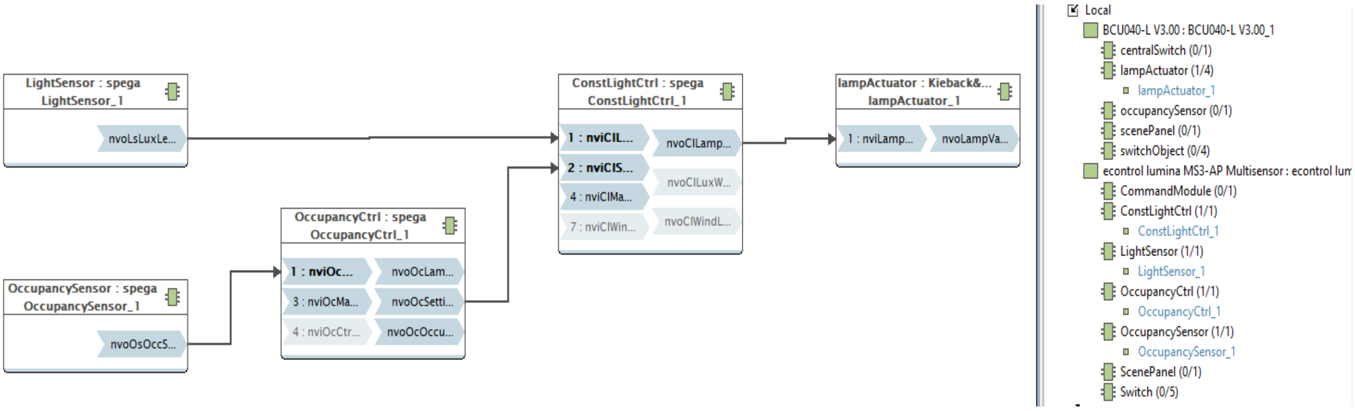
- Search for the device function and the corresponding device that realize a specific abstract function;
- Search for devices and their functions realizing specific requirements;
- Search for all requirements that a device partly or fully implements.

## 5.2 Retrofitting and extension of existing RAS

During the lifecycle of a RAS, there are many scenarios that could require adaptation or extension of existing functionalities. Examples are:

- During the retrofitting design process of a RAS, if there are some automation devices installed in the building, they will be taken into account in order to reduce the acquisition cost of the whole project;
- In large modern residential buildings, the same basic equipment kit of automation devices covering the main needs of users will be installed in all apartments. When the inhabitants will be moved in, they'll have a chance to define their user-specific requirements for automation functions, or simply specify the abstract design with their hardware-related requirements such as technologies' demand. In any case, the basic requirement kit needs to be extended to cover any new requirements.
- During the runtime of RAS, the usage of different locations within the building might be changed according to the current need of users. For example, a study room in the apartment of a single student might need to be changed into a child's room when the owner has a child. In this case the automation systems have to adapt to fulfil the new usage of this room.

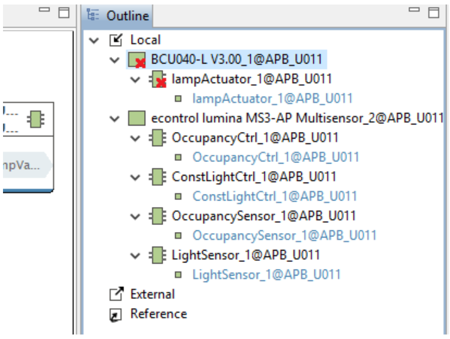
The renovation scenarios (defined in WP2 and WP3) were analysed in order to identify those necessary activities to be done by the planer. The AUTERASsuite was enhanced with a new GUI and back-end functionalities to allow computer-aided support for the planer with these activities.



☑ Local
BCU040-L V3.00 : BCU040-L V3.00_1
centralSwitch (0/1)
lampActuator (1/4)
lampActuator_1
occupancySensor (0/1)
scenePanel (0/1)
switchObject (0/4)
econtrol lumina MS3-AP Multisensor : econtrol lum
CommandModule (0/1)
ConstLightCtrl (1/1)
ConstLightCtrl_1
LightSensor (1/1)
LightSensor_1
OccupancyCtrl (1/1)
OccupancyCtrl_1
OccupancySensor (1/1)
OccupancySensor_1
ScenePanel (0/1)
Switch (0/5)

**Figure 21: Example of installed RAS functions**

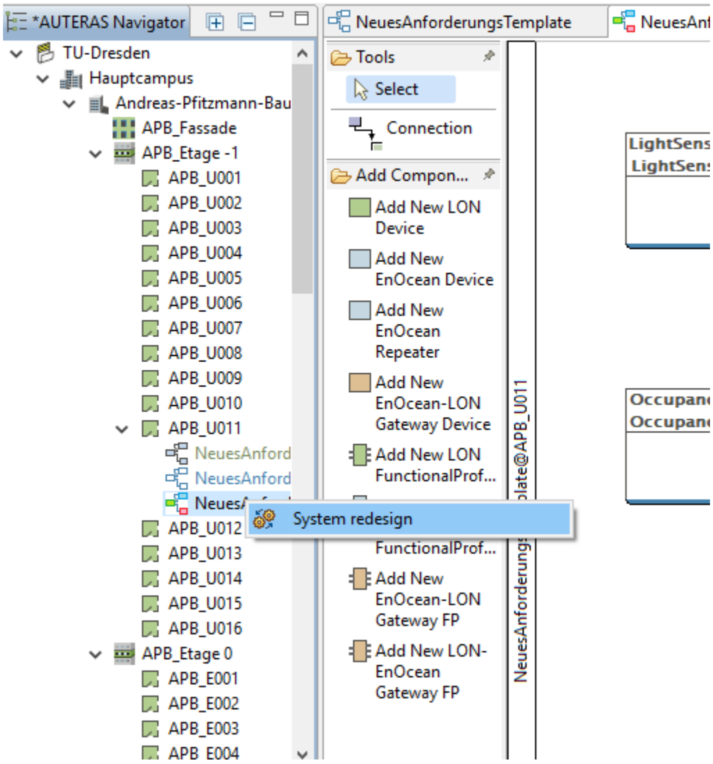
The new functions will be illustrated through the following example. The starting point is a RAS that was already designed, the example is shown in Figure 21. This current system realizes a constant lighting functionality. The system' model was fully created, and it is available for retrofitting processes. From this BAC the lamp actuator should be changed to lower energy consumption of the light. The user has to mark devices which should be changed (see Figure 22).



Outline
Local
BCU040-L V3.00_1@APB_U011
lampActuator_1@APB_U011
lampActuator_1@APB_U011
econtrol lumina MS3-AP Multisensor_2@APB_U011
OccupancyCtrl_1@APB_U011
OccupancyCtrl_1@APB_U011
ConstLightCtrl_1@APB_U011
ConstLightCtrl_1@APB_U011
OccupancySensor_1@APB_U011
OccupancySensor_1@APB_U011
LightSensor_1@APB_U011
LightSensor_1@APB_U011
External
Reference

**Figure 22: Marking device to change during the retrofit process**

The AUTERAS' user can trigger the retrofitting process with the design template directly as it shown in Figure 23 below:



*AUTERAS Navigator
TU-Dresden
Hauptcampus
Andreas-Pfitzmann-Bau
APB_Fassade
APB_Etage -1
APB_U001
APB_U002
APB_U003
APB_U004
APB_U005
APB_U006
APB_U007
APB_U008
APB_U009
APB_U010
APB_U011
*NeuesAnford
NeuesAnford
NeuesA
APB_U012
APB_U013
APB_U014
APB_U015
APB_U016
APB_Etage 0
APB_E001
APB_E002
APB_E003
APB_E004

NeuesAnforderungsTemplate
NeuesAnf
Tools
Select
Connection
Add Compon...
Add New LON Device
Add New EnOcean Device
Add New EnOcean Repeater
Add New EnOcean-LON Gateway Device
Add New LON FunctionalProf...
FunctionalProf...
Add New EnOcean-LON Gateway FP
Add New LON-EnOcean Gateway FP
System redesign
NeuesAnforderungs...plate@APB_U011
LightSens
LightSens
Occupan
Occupan

**Figure 23: Trigger redesign process of a room automation system**

The abstract design (RAS-schematics) of the system will be loaded from the database. In the first step of this process, the information about devices implementing each abstract function block will be retrieved using the mapping between objects of abstract and detailed design templates. This information will be defined as requirements for the current function blocks in the abstract template. After the retrofitting process those function blocks which are already implemented by existing devices will still be realized by the same devices. An exception to this is the information regarding the device that has to be changed.

The device selection will be triggered again taking the abovementioned requirements for the individual function blocks into consideration. Several results are presented to the user, one of which will be selected by the user as the new design for the retrofitted location. An example of the resultant detailed design template is shown in Figure 24.

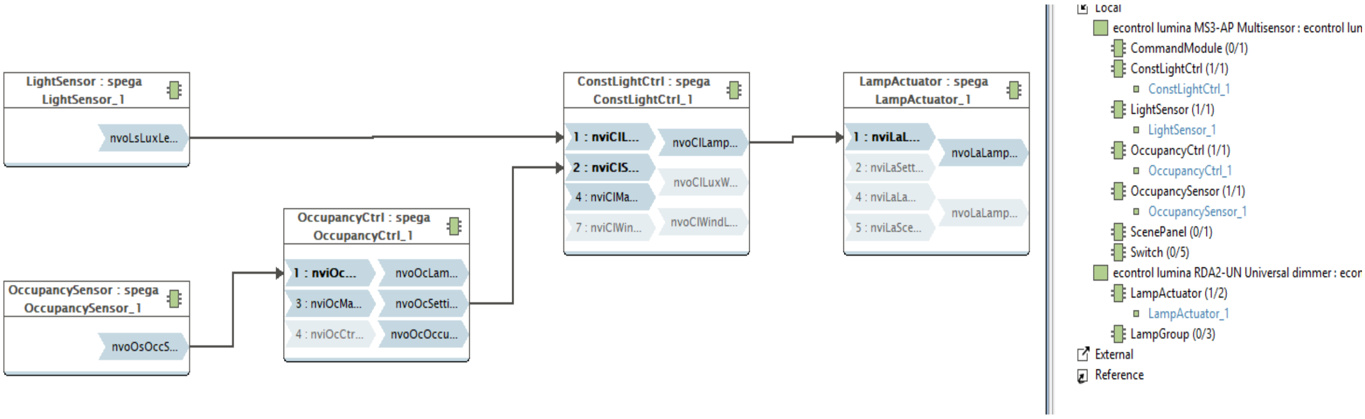


**Figure 24: Resulting detailed design after the retrofit**

As seen in Figure 24, the new device has been selected to realize the lamp actuator function.

The same technology can be used to extend the RAS with new functionalities. This can be done by extending the abstract design before the retrofit process. In the case where newly added function blocks can also be implemented by the existing devices before retrofitting (using the free functional capacity of corresponding devices, means functions from the device that are not used in the current state of the RAS), the existing devices can be reused for these new purposes. This can be done as part of the device selection process. Designs that reuse existing devices are preferable than using other design candidates. This is particularly true when considering criteria such as device numbers or acquisition cost. Therefore, they have better chances to progress during the post-processing phase of the device selection process.

The current RAS model is adapted to the actual state, including mapping between the objects on different model layers.

# 6 Conclusions

The main goals of the work done in T6.2 of the BIM4EEB project was to research and develop supporting tools for stakeholders involved in the post-procurement planning process of RAS to be renovated.

As an initial point of development, the planning tool AUTERAS was taken by the TUD. This tool suite was initially designed and implemented to plan new building automation and control systems.

Furthermore, to match the WP6 expectations, this tool was enhanced and extended with the following main features:

- Integration of a new web service-based IFC-importer to receive the initial building structure automatically from BIMMS (developed in WP4). This allows direct re-use of data which was collected formerly by other partners;
- Replacement of the existing device search algorithm of the AUTERASsuite and therefore an enhancement of its capabilities in the design process of building automation and control systems;
- Researching and implementing of the new methodology in the AUTERASsuite for coping with retrofit scenarios. This allows the design process to handle the existing RAS, which should be enhanced or partially replaced.

The tool was tested via loading the Italian and Polish test sites' structures into the AUTERASsuite via the new importer from BIMMS. After that, new room automation systems were designed and assigned to the different rooms of the test sites. Finally, these automation systems were redesigned (different devices were replaced with other ones) and stored into the AUTERASstore. This completes the workflow of the AUTERASsuite.

BIM4EEB now offers a strong reliable component for the user-support in planning processes. The results of planning processes could be stored in the system information model (described in D6.1) and re-used in T6.3 for checking further building design constraints.

As mentioned in the beginning of this document, the AUTERAS is implemented as the client-server-application. The main reason is that algorithms are very CPU- and memory-intensive. Therefore, all algorithms of the AUTERAS are running on a server at TUD. The complex GUI (AUTERASsuite) is also available for explaining the use of these functionalities.

As the AUTERAS is currently not a complete market-ready tool but an extended system prototype, it is available for project partners for testing if needed. This GUI can be simply downloaded from the AUTERAS web site (www.auteras.de) for installation (approx. 140MB needed on hard disk). Only an up-to-date JAVA-virtual-machine and the certificate (from TUD for security reasons) are needed. For non-experts in the field of BAC, e.g. occupants or end-users, the web-GUI of AUTERASlite is also available on the web site.

All results from work with the AUTERAS can be stored in the system information model (based on AUTERASstore) for further use by other BIM4EEB tools.

## 7 Bibliography

ASHRAE. (2019). *ASHRAE Handbook - HVAC Applications.* Atlanta: ASHRAE.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms* (1 ed.). John Wiley & Sons Ltd.

Kleiber, W., Simon, J., & Weyers, G. (1998). Verkehrswertermittlung von Grundstücken. *Bundesanzeiger*, 2123.

Lehmann, M., Schäfer, F., Sattler, J., & Kabitzsch, K. (2018). Complexity of Design Space Exploration for Building Automation Systems and Comparison of Designing Strategies. *ETFA - International Conference on Emerging Technologies and Factory Automation*.

Özlük, C. (2013). *Design Space Exploration for Building Automation Systems* (1 ed.). Dresden: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-130600.

Ploennigs, J., Lehmann, M., Wollschlaeger, B., Mai, T. L., & Kabitzsch, K. (2014). Multi-objective Device Selection Approach for Component-based Automation Systems. *ETFA - International Conference on Emerging Technologies and Factory Automation*.

TOPAs. (2015). *Tools for Continuous Building Performance Auditing*. (H2020 – grant agreement No 676760) Retrieved from https://www.topas-eeb.eu/

VDI3813:2011-05. (2011). *Building Automation and Control Systems (BACS) standard.* Retrieved from http://bms-technology.com/pdf/VDI_3813-1.pdf