



# BaaS

## Building as a Service

**FP7-ICT-2011-6: ICT Systems for Energy Efficiency  
Small or Medium-scale Focused Research Project  
Grant Agreement No. 288409**

### **Deliverable D4.1:**

## **Simulation and modelling tools available for the model Software**

Deliverable Version:	D4.1, v.3.0
Document Identifier:	BaaS_WP4_D4.1_SimulationModellingTools_1.0
Preparation Date:	January 13, 2014
Document Status:	Final (Resubmission)
Author(s):	M A García <sup>3</sup> , K Höttes <sup>4</sup> , G I Giannakis <sup>2</sup> , G Kontes <sup>2</sup> , G N Lilis <sup>2</sup> , J Rojicek <sup>1</sup> , D Rovas <sup>2</sup> , J R Santiago <sup>4</sup> , C Valmaseda <sup>3</sup> ( <sup>1</sup> HON, <sup>2</sup> TUC, <sup>3</sup> CARTIF, <sup>4</sup> Fraunhofer)
Dissemination Level:	CO - Confidential, only for members of the consortium (including the Commission Services)



Project funded by the European Community  
in the 7<sup>th</sup> Framework Programme



ICT for Sustainable Growth





## Deliverable Summary Sheet

### Deliverable Details

<b>Type of Document:</b>	Deliverable
<b>Document Reference #:</b>	D4.1
<b>Title:</b>	Simulation and modelling tools available for the model Software
<b>Version Number:</b>	3.0
<b>Preparation Date:</b>	January 13, 2014
<b>Delivery Date:</b>	January 13, 2014
<b>Author(s):</b>	M A García <sup>3</sup> , K Höttinges <sup>4</sup> , G I Giannakis <sup>2</sup> , G Kontes <sup>2</sup> , G N Lilis <sup>2</sup> , J Rojicek <sup>1</sup> , D Rovas <sup>2</sup> , J R Santiago <sup>4</sup> , C Valmaseda <sup>3</sup> ( <sup>1</sup> HON, <sup>2</sup> TUC, <sup>3</sup> CARTIF, <sup>4</sup> Fraunhofer)
<b>Document Identifier:</b>	baas_wp4_d4.1_simulationmodellingtools_2.0.docx
<b>Document Status:</b>	Final (Resubmission)
<b>Dissemination Level:</b>	CO - Confidential, only for members of the consortium (including the Commission Services)

### Project Details

<b>Project Acronym:</b>	BaaS
<b>Project Title:</b>	Building as a Service
<b>Project Number:</b>	288409
<b>Call Identifier:</b>	FP7-ICT-2011-6
<b>Call Theme:</b>	ICT Systems for Energy Efficiency
<b>Project Coordinator:</b>	Fundacion Cartif (CARTIF)
<b>Participating Partners:</b>	<p>NEC Europe Ltd. (NEC, UK);</p> <p>University of Ireland, Cork (UCC-IRU (IE)</p> <p>Dalkia Energia y Servicios (DALKIA, ES)</p> <p>Honeywell, spol. s r. o. (HON, CZ);</p> <p>Technical University of Crete (TUC, GR);</p> <p>Fraunhofer-Gesellschaft zur Förderung der Angewandten Forschung e.V. (Fraunhofer, DE);</p> <p>Foundation Cartif (CARTIF, ES);</p>
<b>Instrument:</b>	STREP
<b>Contract Start Date:</b>	May 1, 2012
<b>Duration:</b>	36 Months

### Deliverable D4.1: Short Description

The aim of this deliverable is to define a landscape and requirements from modelling services. It sets the foundation for supportive APO services which are consumed by other APO services delivering insights to BaaS users.

**Keywords:** simulation; prediction; APO.

### Deliverable D4.1: Revision History

Version:	Date:	Status:	Comments
0.1	30/10/2012	Initial	Fraunhofer: Initial schema for the deliverable. ZUB Building case description (previously named v1.1)
0.2	22/11/2012	Draft	CARTIF: schema revision (previously named v1.1_01)
0.3	30/11/2012	Draft	CARTIF: Cartif 1 offices building description (previously named v1.1_02)
0.4	30/04/2013	Draft	CARTIF: new structure and draft using D5.1
0.5	08/05/2013	Draft	Fraunhofer: section 4.5 added
0.6	28/05/2013	Draft	Fraunhofer: review of section 4.5 (new: 4.5.1 to .6, .8, .10.3)
0.7	07/06/2013	Draft, IR	Fraunhofer: Intermediate Report (IR)
0.8	27/08/2013	Draft	Fraunhofer: review of data acquisition and appendix
0.8	10/09/2013	Draft	TUC: review of sections regarding simulation, requirements and data
0.9	10/09/2013	Draft	Fraunhofer: review
0.10	12/09/2013	Draft	all: internal review
1.0	13/09/2013	Final	CARTIF: final review and submission
2.0	28/11/2013	Draft	Fraunhofer: new version of the deliverable according to the Review Report
2.01-2.10	31/12/2013	Draft versions	TUC, CARTIF and Fraunhofer: modifications in the content in order to complete the sections related to requirements, MVDs and simulation model requirements for demo sites
2.14	10/01/2014	Internal review	Final version for internal review
3.0	13/01/2014	Final	Final version for resubmission

### Copyright notices

© 2014 BaaS Consortium Partners. All rights reserved. BaaS is an FP7 Project supported by the European Commission under contract #288409. For more information on the project, its partners, and contributors please see <http://www.baas-project.eu/>. You are permitted to copy and distribute verbatim copies of this document, containing this copyright notice, but modifying this document is not allowed. All contents are reserved by default and may not be disclosed to third



---

parties without the written consent of the BaaS partners, except as mandated by the European Commission contract, for reviewing and dissemination purposes. All trademarks and other rights on third party products mentioned in this document are acknowledged and owned by the respective holders. The information contained in this document represents the views of BaaS members as of the date they are published. The BaaS consortium does not guarantee that any information contained herein is error-free, or up to date, nor makes warranties, express, implied, or statutory, by publishing this document.



## Table of Contents

1	Introduction.....	2
1.1	Purpose and target group .....	2
1.2	Contribution of partners .....	2
1.3	State of the Art.....	3
1.4	Relation to other activities in the project.....	3
2	Simulation for energy performance estimation.....	4
2.1	Calculation Methodologies – Temporal Classification.....	4
2.1.1	<i>Quasi-static Calculation Methodologies.....</i>	<i>5</i>
2.1.2	<i>Transient Calculation Methodologies.....</i>	<i>7</i>
2.1.3	<i>State-space calculation methodologies.....</i>	<i>13</i>
2.1.4	<i>Uncertainties and Validation.....</i>	<i>14</i>
2.2	Simulation Engines and Co-Simulation .....	15
2.2.1	<i>Whole Building Thermal Simulation Engines.....</i>	<i>16</i>
2.2.2	<i>Co-Simulation.....</i>	<i>19</i>
3	Simulation task in BaaS and its functional requirements.....	21
3.1	Model type requirements.....	24
3.1.1	<i>Calculation Methodologies to be supported in BaaS.....</i>	<i>24</i>
3.1.2	<i>Simulation models.....</i>	<i>26</i>
3.2	Input Data Requirements.....	29
3.2.1	<i>Static data.....</i>	<i>29</i>
3.2.2	<i>Dynamic data.....</i>	<i>34</i>
3.2.3	<i>Access to dynamic data: co-simulation.....</i>	<i>37</i>
3.2.4	<i>Conclusions.....</i>	<i>38</i>
3.3	APO layer integration requirements.....	39
3.3.1	<i>Data flow.....</i>	<i>39</i>
3.3.2	<i>Integrating simulation capabilities as services to the APO layer.....</i>	<i>40</i>
3.3.3	<i>Simulation Manager.....</i>	<i>41</i>
3.3.4	<i>Configuration.....</i>	<i>42</i>
4	Model View Definitions (MVDs) for thermal simulation.....	43
4.1	Method for defining and developing interfaces - introduction.....	43
4.2	Model View Definitions .....	43
4.3	Definition of "exchange requirements" .....	44
5	Data acquisition.....	46
5.1	Actual state of versions and tools (summary).....	46

---

5.2	Sources of information, references .....	47
5.3	IFC-structure regarding thermal simulation .....	48
5.3.1	<i>IFC schemas and domains</i> .....	48
5.3.2	<i>IFC data types</i> .....	49
5.4	Simulation model development .....	51
5.5	Modules for simulation models of subsystems .....	60
5.6	Transferability of simulation modules on pilot buildings .....	60
5.7	Example for the application of the generic simulation modules on the pilots ..	61
5.7.1	<i>Abstract simulation model approach</i> .....	61
5.7.2	<i>Main simulation model</i> .....	62
5.7.3	<i>Macro Generation Control</i> .....	62
5.7.4	<i>Macros zones</i> .....	63
5.7.5	<i>Macro Building + Weather</i> .....	64
5.8	Model configuration .....	65
5.9	Mapping the IFC file .....	66
5.10	Relation of simulation component parameters and data sources .....	67
5.10.1	<i>Generation of heat and cold (water)</i> .....	67
5.10.2	<i>Flow diverter</i> .....	69
5.10.3	<i>Flow mixer</i> .....	70
5.10.4	<i>Pump</i> .....	71
5.10.5	<i>Pipe</i> .....	72
5.10.6	<i>Single zone (radiators)</i> .....	73
5.10.7	<i>Building model (Type 56)</i> .....	75
5.10.8	<i>Co-simulation configuration</i> .....	77
6	Requirements for thermal simulation models for demo sites .....	82
7	References .....	86
	Appendix A: Querying of IFC models and sample query .....	89



## List of Figures

Figure 1: Schematic of the ISO 13790:2008 calculation methodology [16] .....	6
Figure 2: 5RIC Representation of a zone [16] .....	7
Figure 3: TRNSYS 17 Simulation Model for the FIBP Test-Bed (left); EnergyPlus Simulation Model for the TUC Test-bed (right) .....	8
Figure 4: Zoning of the TUC building (above); connectivity graph for the thermal interactions (below) ...	9
Figure 5: Zoning of the TUC building to account correctly for solar gains through the atrium (top); connectivity graph for inter-zone thermal interactions (bottom). .....	10
Figure 6: System of Resistances and Capacitances (SRC) .....	11
Figure 7: CFD calculations for the TUC test-bed: temperature distribution in a zone during heating (left); external CFD calculation to determine the pressure and velocity fields. ....	12
Figure 8: Calculation Methodologies – Input-Output Relationship .....	14
Figure 9: TRNSYS simple HVAC example .....	20
Figure 10: Amalgamation of discussion in this section .....	25
Figure 12: Projections of a building element and overlapping of two building elements .....	32
Figure 13: Angled wall connection.....	32
Figure 14: Example of construction of building elements.....	33
Figure 15: Interconnection principle, i.e. WP4, WP5 and WP6 .....	40
Figure 16: Data exchange via co-simulation .....	40
Figure 18: IFC domains and contribution of partners.....	49
Figure 19: Simple model of boiler/pump/radiator circuit (IFC viewer) .....	51
Figure 21: Advanced systems (TRNSYS) - parallel demand .....	52
Figure 22: Advanced system - multiple generators .....	53
Figure 23: Cross Sectional Schematic of a Two-Pipe Console Unit.....	55
Figure 24: Example simulation model (TRNSYS) of air handling unit .....	56
Figure 25: Air handling unit and sub-components - scheme .....	57
Figure 26: Air handling unit and sub-components - scheme and corresponding TRNSYS model (simplified).....	57
Figure 27: Air handling unit and sub-components - connection to zones.....	58
Figure 28: Generic simulation model - heating system module .....	60
Figure 29: Generic simulation model - cooling system module .....	60
Figure 30: Abstract scheme of heating system .....	61
Figure 31: Entire simulation model of Sierra Elvira School (TRNSYS).....	62
Figure 33: Content of macro "Macros Zones" .....	64
Figure 35: Different principles and samples for the mapping process.....	67
Figure 36: Screenshot SketchUp model of school (IDF file) .....	77
Figure 38: Screenshot SketchUp model of school - two zones .....	79

---

Figure 39: Boiler circuit with 3-way valve and tee piece .....	90
Figure 40: Screenshot of Java tool during execution.....	92
Figure 41: Screenshot of IFC-file in 'Solibri': values of PropertySet "Dimensions" .....	96
Figure 42: Screenshot of Java tool during execution - generation of simulation input file .....	98
Figure 43: Screenshot of editor - simulation input file .....	99
Figure 44: Screenshot of simulation engine .....	100
Figure 45: Screenshot of simulation result .....	100
Figure 46: Boiler circuit and relations .....	101

---

## List of Tables

Table 1: Summary of Contributions of Partners .....	2
Table 2: D4.1 tasks relationship with other BaaS work packages .....	3
Table 3: APO services model requirements .....	22
Table 4: Describing the differences between 1 <sup>st</sup> (left) and 2 <sup>nd</sup> (right) level space boundaries – TUC building example.....	31
Table 5: APO tasks and their simulation functional requirements.....	38
Table 6: Simulation Data Requirements and their data sources.....	39
Table 7: Sources of information, references .....	47
Table 8: List of relevant Entities - HVAC.....	50
Table 9: List of relevant Entities - Control .....	50
Table 10: List of relevant Entities - Electricity.....	51
Table 11: TRNSYS types for components in heating and cooling circuits.....	54
Table 12: TRNSYS types for components in AHU .....	58
Table 13: Thermal simulation models for the pilot buildings and test-beds .....	61
Table 14: Simulation parameter obtained from IFC, data warehouse DWH and BaaS system for generic heating circuit.....	65
Table 15: Parameter, Inputs and Outputs – Waterboiler .....	68
Table 16: Parameter, Inputs and Outputs - Flow diverter .....	70
Table 17: Parameter, Inputs and Outputs - Flow mixer .....	70
Table 18: Parameter, Inputs and Outputs – Pump.....	71
Table 19: Parameter, Inputs and Outputs – Pipe.....	72
Table 20: Parameter, Inputs and Outputs - Single zone (radiators) .....	74
Table 21: Parameter, Inputs and Outputs - Building model.....	75
Table 22: Summary of data exchange for the example.....	80
Table 23: Prediction model requirements .....	82

---

## Abbreviations and Acronyms

<b>AEC</b>	Architecture, Engineering and Construction
<b>APO</b>	Assess, Predict, Optimize
<b>ASHRAE</b>	American Society of Heating, Refrigerating and Air-Conditioning Engineers
<b>BaaS</b>	Building as a Service
<b>BACnet</b>	Building Automation and Control Network
<b>BACS</b>	Building Automation and Control Systems
<b>BAN</b>	Building Area Network
<b>BIM</b>	Building Information Model
<b>BIMServer</b>	TNO Open Source Building Information Model Server
<b>BMS</b>	Building Management System
<b>CAO</b>	Cognitive Adaptive Optimization
<b>CC</b>	Core Communication
<b>CL</b>	Communication Logic
<b>CoP</b>	Coefficient of Performance
<b>CSV</b>	Comma-Separated Values (a universal file format for tabular data)
<b>DACM</b>	Data Acquisition and Control Manager
<b>DAO</b>	Data Access Object
<b>DB</b>	Database
<b>DBMS</b>	Database Management System
<b>DC</b>	Domain Controller
<b>DoW</b>	Description of Work
<b>DW</b>	Data Warehouse
<b>eBIM</b>	Extended Building Information Model
<b>E2B</b>	Energy Efficient Buildings
<b>ESCO</b>	Energy Service Company
<b>ETL</b>	Extract, Transform, Load
<b>FD</b>	Fault Detection
<b>FDD</b>	Fault Detection and Diagnosis
<b>FDI</b>	Fault Detection and Identification
<b>FM</b>	Facility Management
<b>gbXML</b>	Green Building XML

---

<b>HESMOS</b>	ICT Platform for Holistic Energy Efficiency Simulation and Lifecycle Management of Public Use Facilities
<b>HVAC</b>	Heating, Ventilation, and Air Conditioning
<b>ICT</b>	Information and Communication Technology
<b>IDM</b>	Information Delivery Manual
<b>IEC</b>	International Electro-technical Commission
<b>IFC</b>	Industry Foundation Class
<b>intUBE</b>	Intelligent Use of Building Energy Information
<b>ISO</b>	International Organization for Standardization
<b>JDBC</b>	Java Database Connectivity
<b>KPI</b>	Key Performance Indicator
<b>MVD</b>	Model View Definition
<b>ODBC</b>	Open Database Connectivity
<b>OLAP</b>	On-Line Analytical Processing
<b>OSGi</b>	Open Services Gateway Initiative
<b>PEBBLE</b>	Positive-Energy Buildings through Better control dEcisions
<b>PPD</b>	Predicted Percentage Dissatisfied
<b>PMV</b>	Predicted Mean Vote
<b>REST</b>	REpresentational State Transfer
<b>RPC</b>	Remote Procedure Call
<b>SOA</b>	Service-Oriented Architecture
<b>SOAP</b>	Simple Object Access Protocol
<b>SOTA</b>	State of the Art
<b>TC</b>	Thermal Comfort
<b>TNO</b>	Netherlands Organization for Applied Scientific Research
<b>VC</b>	Visual Comfort
<b>WP</b>	Work Package
<b>WS</b>	Web Service
<b>WSDL</b>	Web Service Definition Language
<b>XML</b>	Extensible Markup Language

---



## Executive Summary

The aim of this deliverable is to define a landscape and requirements from modelling services. It sets the foundation for supportive APO services which are consumed by other APO services delivering insights to BaaS users (for further information see D5.1 – Functional and interoperability requirements for building services). The reader should understand how modelling capability is used in BaaS platform, what its purpose is and how it interfaces with other BaaS parts. Unlike originally envisioned, modelling services do not communicate directly with the warehouse through the middleware layer – it was found more natural that modelling is understood as supportive APO service which results can be consumed by any other service. This has a good reason: end user is rarely directly interested in simulation results only as such results do not bring any actionable message; the results should always be accompanied by results interpretation which is a goal of other APO services. This approach also makes easier the integration of simulation tools into BaaS platform infrastructure.

This should not be understood as modelling tools have no overlap with BIM, warehouse and other external systems as indicated in DoW: on contrary modelling tools heavily depends on such data, so APO kernel as the data interface between data sources and modelling tool must take into account all need of the modelling services.

More specifically, this deliverable addresses functional requirement of simulation tasks required underlying Model View Definitions for thermal simulation and also data acquisition needs.

# 1 Introduction

The goal of this deliverable is to explain the role of simulation services in BaaS platform that means to connect known building optimization use cases and their addressing APOs to simulations. This deliverable should be read along with D5.1 for understanding BaaS approach from a perspective.

The main target of the revision of this deliverable bases on the parallel revision of deliverable D5.1 and the preparation of deliverable D4.2. As an outcome of discussions within the consortium it became clear to merge the effort in all three deliverables but to separate more clearly the different contents. Thus, this deliverable D4.1 focuses on the development of generic modules of simulation models in order to enable the interconnection to the BaaS system, while deliverable D4.2 describes the development of simulation models regarding the modelling of the BaaS demo side buildings (pilots).

## 1.1 Purpose and target group

The purpose of BaaS deliverable 4.1 is to specify the role of the simulation and modelling tools for the BaaS system. The simulation task and the functional requirements are explained, that means different calculation approaches and methodologies, classification of input data and requirements as well as the integration of the simulation capabilities as supportive services to the APO layer.

An extra attention is paid to transferability of simulation modules among BaaS pilots and possibly other BaaS sites: the deliverable therefore describes generic simulation modules for different subsystems of a building permitting such transferability through core simulation models. The model view definition MVD serves as data manager to enable a data exchange among different data sources, which is a key condition for generic simulation models. The description of input and output parameters for simulation models is based on the defined use cases for the test-beds and pilot buildings.

The target group is composed of all consortium partners (CARTIF, NEC, Honeywell, Fraunhofer, TU Crete, UCC-IRU, and DALKIA) and also authors for potential new APO services,

## 1.2 Contribution of partners

This deliverable is led by Fraunhofer with major contributions from CARTIF, TUC and HON consortium members.

The workload is distributed among the mentioned partners as follows:

**Table 1: Summary of Contributions of Partners**

Partner	Deliverable Focus
Fraunhofer	Deliverable Lead, Simulation system technology, Development of components
HON	MVD, Requirements, Development of components
TUC	Interfaces to the APO Kernel, Simulation Manager Component, MVDs



CARTIF	MVD, Requirements, Establishing interactions with WP2 and WP3 components
--------	--

### 1.3 State of the Art

The issue of energy performance of buildings is of great concern to building owners because it translates to building operation cost. To optimize building performance, many advanced monitoring and optimization tools rely on accurate energy performance modelling – which also applies to many BaaS’ APOs. The level of complexity of building modelling differs depending on actual needs: many tasks can be addressed by data-centric approaches, i.e. by models based on empirical relationships, but in some cases it is necessary to develop a model based on physical/first principles – the model being more accurate and requiring fewer/weaker assumptions. Within BaaS depending on the actual needs of the APO at hand either of the two approaches may be utilized.

Empirical models are defined by usually a simple formula derived from available data, often commissioning data, capturing only the basic dependencies. First principle models are based on physical dependencies with detailed description of the modelled building and solving a set of differential equations to simulate the real building behaviour. As this task is far from trivial, specialized software like TRNSYS is being used to process all input data. The trade-off is obvious: for more accurate results much more input data must be provided and calculations are more computationally demanding, therefore it is imperative to carefully consider which data are available for any specific task and what is the required level of accuracy: one-size-fits-all does not apply here.

### 1.4 Relation to other activities in the project

This deliverable lays the foundations for the work in several other BaaS work packages. The following table is a brief summary of such relationships.

**Table 2: D4.1 tasks relationship with other BaaS work packages**

D4.1 Section	Relationship
2.3.2 Co-simulation	D5.1: Prediction tasks - simulation
3	D5.1: Assess, Predict, Optimize (APO) tasks in buildings
5	D2.1: extended BIM Specification D5.1: Prediction tasks - simulation
6	D5.1: Requirements (inputs/outputs) for thermal simulation models for demo sites

## 2 Simulation for energy performance estimation

### 2.1 Calculation Methodologies – Temporal Classification

In general model-based building thermal and energy simulation programs use mass and energy balances [14] as a basis for estimation of the evolution of the values of parameters referring to internal conditions (temperature, humidity, CO<sub>2</sub> concentration, luminance) and energy needs (total energy, maximum power demands) of building interiors. Energy conservation laws are used to investigate thermal energy transfers and exchanges among building elements, spaces, and systems, while mass conservation is used for evaluation of vapour-water transfers (humidity). Implicit in all the methodologies is the discretization of the pertinent conservation equations over pre-determined time intervals. Based on time-resolution criteria calculation methodologies can be classified into two categories:

- Static or quasi-static calculation methodologies.

These methods assume average parameter values for a long period of time (typically a month or a season), and account for dynamic effects using empirical correlations and averaging correction factors. These types of calculation methods are especially useful for estimation on energy performance on an annual basis.

- Transient calculation methodologies.

Transient calculation methodologies take a more granular approach using a time resolution which is comparable to the time-scale of time-varying effects that are being modelled. Consequently these methods are capable of capturing transient phenomena such as weather changes, occupancy variations, thermal loading effects, or the effects of building energy management systems.

The monthly-based calculation methodology described in ISO 13790:2008 is a prime example of a quasi-static calculation methodology. This fully-prescribed<sup>1</sup> calculation methodology has been adapted – in the context of activities for the implementation of the EPBD [17] – by many EU member states to form at a national level an accepted calculation methodology for computing energy performance. In Annex H of the standard the accuracy of the calculation methodology and the sensitivity to errors in the input data is discussed. In certain conditions, the calculation methodology can be validated against reality and relatively small deviations can be observed for annual predictions, but on the monthly scale these deviations can be significant. The sensitivity to input data is also discussed: uncertainties in the estimation of thermal properties or other input parameters can contaminate the results, and the propagation of these errors can yield sizable deviations in the end results. For this reason, in many cases, the calculation methodology is used to establish an ordering relation, that allows for meaningful comparisons<sup>2</sup> (and thus establishing the rating system used in many countries), but with lesser expectations with regards to prognosis of real performance.

<sup>1</sup> and to a certain extent “unambiguous,”

<sup>2</sup> e.g., the concept of a reference building

### 2.1.1 *Quasi-static Calculation Methodologies*

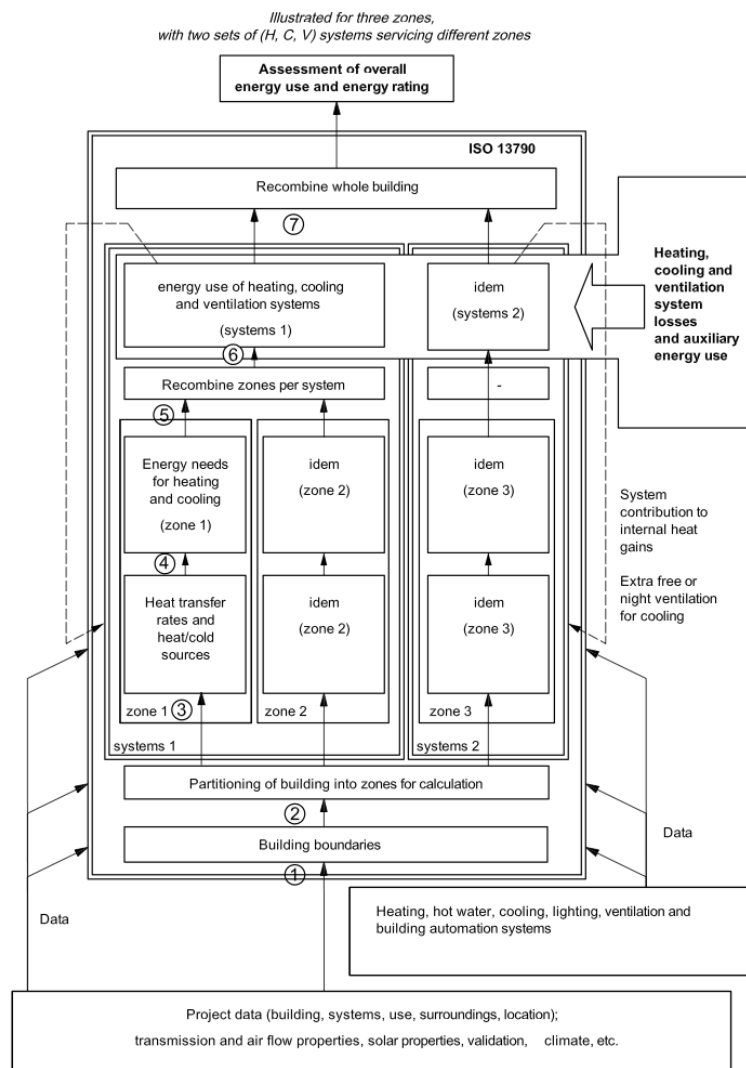
A basic modelling assumption used here is the multi-zonal paradigm: dividing the volume of the building into disjoint regions (zones), each with the basic variables (say, temperature) assumed to be spatially constant. The evolution in time of the zonal parameters is evaluated from the solution of a system or algebraic and/or ordinary differential equations. Zones in that sense form the basic spatial component for performing the calculations. The selection of zoning is in essence the spatial discretization of the building. As it typical in other domains (e.g. the numerical solution of partial differential equations), the resolution should be granular enough to be able to discern all basic effects that are being modelled. So, in effect, the calculation methodology, and in particular the level of modelling detail employed by that methodology, govern the spatial discretization to be used. As we will see later, for more detailed modelling approaches, as the ones used in simulation software like TRNSYS and EnergyPlus, the approach taken to zoning should be different to conform to the higher level of modelling detail employed. Implicit in the choice of discretization is the balance between accuracy and complexity<sup>3</sup>. An overly fine discretization can lead to many input requirements and increase disproportionately the effort required in setting up the calculation. Too coarse of an approximation, implies low complexity but also large approximation errors might be introduced invalidating the obtained results. Obviously selecting the zoning to strike a proper balance between accuracy and complexity is a critical consideration in setting up the simulation, and it is something that cannot be done automatically. In the calculations described in the standards the desire for transparency and reproducibility, push toward coarser zoning definitions – for the quasi-steady calculation methodology discussed so far the interested reader could refer to [18], or other similar standards on a clear set of guidelines. But for more detailed modelling and realistic results, zoning is the single most important assumption that can separate a properly conducted simulation that closely reflects reality from a nonsensical one. The multitude of ways that the spatial discretization can be defined, with the concomitant effects it has with respect to the quality of the simulation, and the ambiguity in its definition esp. for complex building geometries, makes the entering threshold for whole-building simulation quite high. Once the spatial discretization has been established, at a second level the interactions between zones need to be prescribed: in most cases inter-zone exchanges should be enforced, but to simplify the calculation in certain cases (when there is presumed weak thermal coupling) adiabatic boundary conditions can be enforced thus simplifying the calculation. Once the spatial discretization has been established a connectivity graph can be created to represent the interactions between zones.

In Figure 1, a schematic of the calculation methodology for a building split into three zones is illustrated. Quoting the Standard [16], once the zoning has been established, the basic energy interactions that have to be accounted for in forming the energy (heat) balance at the building zone level include the following terms:

- transmission heat transfer between the conditioned space and the external environment, governed by the difference between the temperature of the conditioned zone and the external temperature;
- ventilation heat transfer (by natural ventilation or by a mechanical ventilation system), governed by the difference between the temperature of the conditioned zone and the supply air temperature;

<sup>3</sup> Complexity here refers to the number of inputs that have to be specified, which scale linearly to the number of zones, and also to the computational complexity which also increases as the number of zones increase.

- transmission and ventilation heat transfer between adjacent zones, governed by the difference between the temperature of the conditioned zone and the internal temperature in the adjacent space;
- internal heat gains (including negative gains from heat sinks), for instance from persons, appliances, lighting and heat dissipated in, or absorbed by, heating, cooling, hot water or ventilation systems;
- solar heat gains (which can be direct, e.g. through windows, or indirect, e.g. via absorption in opaque building elements);
- storage of heat in, or release of stored heat from, the mass of the building;
- energy need for heating: if the zone is heated, a heating system supplies heat in order to raise the internal temperature to the required minimum level (the set-point for heating);
- energy need for cooling: if the zone is cooled, a cooling system extracts heat in order to lower the internal temperature to the required maximum level (the set-point for cooling).



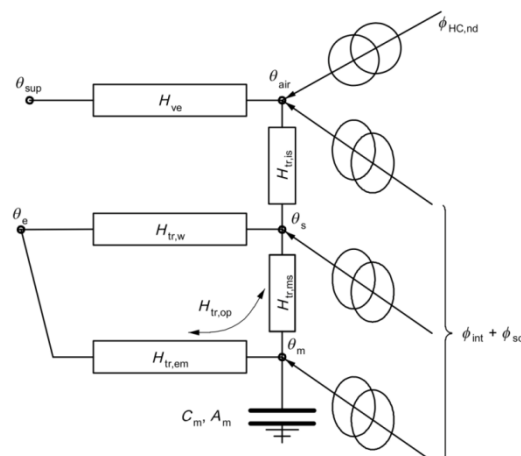
**Figure 1: Schematic of the ISO 13790:2008 calculation methodology [16]**

The basic energy calculations involving the terms described before, are performed at each zone and then combined to estimate the energy use for heating, cooling and ventilation systems. The whole process is repeated in an iterative manner. Upon recombination from the zones to the whole building energy-use indices are computed for the whole building. Boundaries to the calculation are the presence of systems for heating, hot water, cooling, lighting, ventilation and building automation systems. Their presence and concomitant calculation methodologies modelling their effects are stipulated in other standards and should be used together with the building model.

### 2.1.2 Transient Calculation Methodologies

In the case where dynamic effects are important the temporal resolution of a month is not sufficient to capture all relevant dynamics. In this case, smaller time steps are required and a different approach is essential. This has obvious benefits: certain physical effects like transfer of heat from building thermal masses or the dynamic effects of the operation of active climate control HVAC components happen on a time-scale which is comparable to the simulation time-step. It is then possible to use more detailed models that capture these dynamic interactions and there is no longer the need for averaging or the use of correlations and other correction factors. On the other hand the need for defining boundary conditions, at each time steps means that in many cases the problem definition has to be more detailed (at each time step) requiring, at this level of detail, information which may not be available.

An example of such model is the dynamic one described in [16]. Here a zone is represented as a thermal circuit with 5 thermal resistances and 1 thermal capacitance (5R1C).



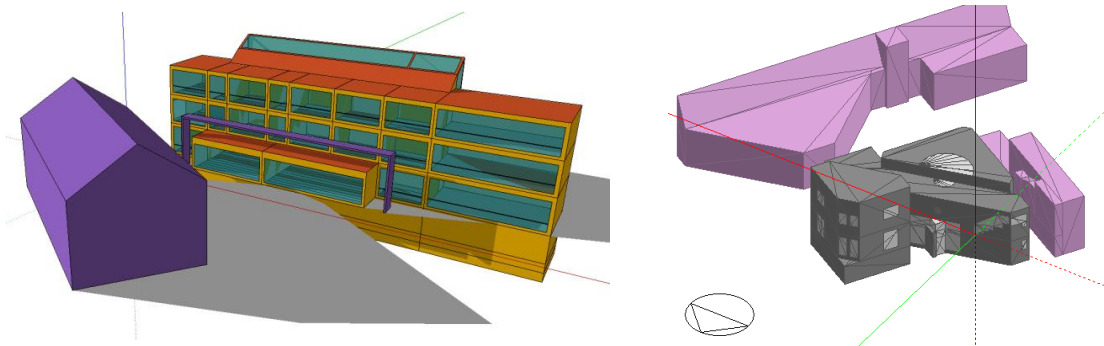
**Figure 2: 5R1C Representation of a zone [16]**

The thermal capacitance models thermal storage effects in the zone. The mathematical formulation of the problem in this case is an Ordinary Differential Equation to model the evolution of the temperature  $\theta_m$  as a function of time. Upon discretization of the equation using a finite-difference scheme, e.g. an implicit scheme like the Crank-Nicolson method, one gets the equations for the evolution in time of the relevant temperature parameter. One obvious benefit in the model above is the ability to model the temperature in the walls and therefore it becomes possible to have estimates of thermal comfort (as the radiant temperature is an important parameter for thermal comfort). In a multi-zone configuration one needs to set individually for each of these nodes the thermal system and combine it to form the overall thermal network. The number of capacitances in this case is proportional to the number of zones, and a system of

Ordinary Differential Equations has to be integrated in time. In ISO 13790 such a methodology is described and the boundary conditions are selected to ensure compatibility between the monthly and dynamic models.

A similar approach is followed in TRNSYS 16<sup>4</sup> [4]: TRNSYS has a modular and extensible structure where different models for the building and its systems (called Types in TRNSYS lingo) are combined to form the problem description. Type 56, implements the multi-zone building model. There the geometrical and connectivity information for the zonal splitting is provided along with parameters for describing opaque and transparent building materials. The models used for the multi-zone building are more detailed than the simple RC above, including a star-shaped topology for approximating radiant exchange between zone surfaces along with the transfer function method for modelling transient conductive exchanges through walls. The integration time step can vary from 1min to 1h<sup>5</sup>. This higher level of simulation detail is especially useful when one considers the coupled interaction of the building and energy systems; for this reason it has been extensively used as a simulation-aid tool for energy systems development and testing.

When predilection for temperatures in the zones or other dynamic effects exists, accurate modelling of thermal gains is essential. This is particularly important in cases of buildings with high solar gains e.g. due to a high glazing-wall ratio or due to the presence of solar atriums.



**Figure 3: TRNSYS 17 Simulation Model for the FIBP Test-Bed (left);  
EnergyPlus Simulation Model for the TUC Test-bed (right)**

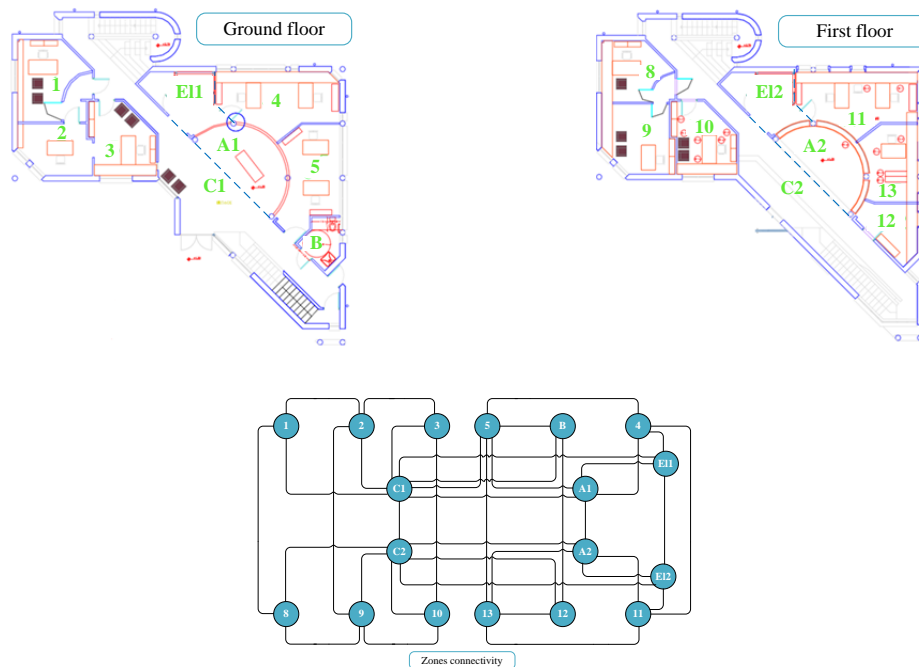
Shown in Figure 3 are the simulation models for two of the BaaS test-beds. On the left, the ZUB building of the Fraunhofer Institute of Building Physics (FIBP) is shown, on the right the Technical Services Building at the Technical University of Crete (TUC). In the case of the FIBP

<sup>4</sup> Here and in the discussion that follows two specific simulation toolboxes are explicitly mentioned: TRNSYS and EnergyPlus. While the examples given here use these tools for specificity reasons, it should be understood that these tools represent only a small fraction of the options available for whole building simulation. Nonetheless, the discussion regarding temporal and spatial discretization options applies irrespective of the actual implementation selected. The discussion here intends to classify simulation approaches and in no way should be construed as endorsement on the part of the BaaS consortium of any of these tools. For the reader interested in exploring other tools, see [9] for a, not necessarily up-to-date, but nonetheless comprehensive comparison.

<sup>5</sup> In fact, the use of double precision numbers, allows for time-steps as low as 0.01 sec, although it might be hard to justify the use of such time-steps as the time scales related to the thermal processes modeled is at least an order of magnitude slower.

building the high glazing to wall ratio along with the proper orientation of the building allows for large solar gains through the triple glazing, effectively reducing heating demands in the winter time. Improper control of external shading devices during summer time can cause overheating due to the large thermal gains. In the case of the TUC building the presence of a solar atrium allows natural light for the building corridors, but causes significant overheating especially during the summer months when the solar path is at an almost perpendicular angle with the roof.

In both cases, to compute correctly solar gains and the evolution of temperature variables, a detailed consideration of the three-dimensional building geometry is necessary. For such reasons, state-of-the-art building simulation software today, uses the 3-D information of the building elements, to define geometrically the building spaces and surfaces, and uses this information to compute in more detail radiative transfer, e.g. due to solar gains. Examples of tools which implement such calculation methodologies are TRNSYS 17 [4] and EnergyPlus [3]. Both of these tools use detailed three-dimensional representation of geometric objects-building elements, and use detailed<sup>6</sup> methods to compute such exchanges. At this level of modelling detail, obvious advantages are the capability to capture correctly radiant heat gains and exchanges. It also becomes possible to include the shading effects of neighbouring buildings, represented as external shading surfaces (shown in Figure 3).

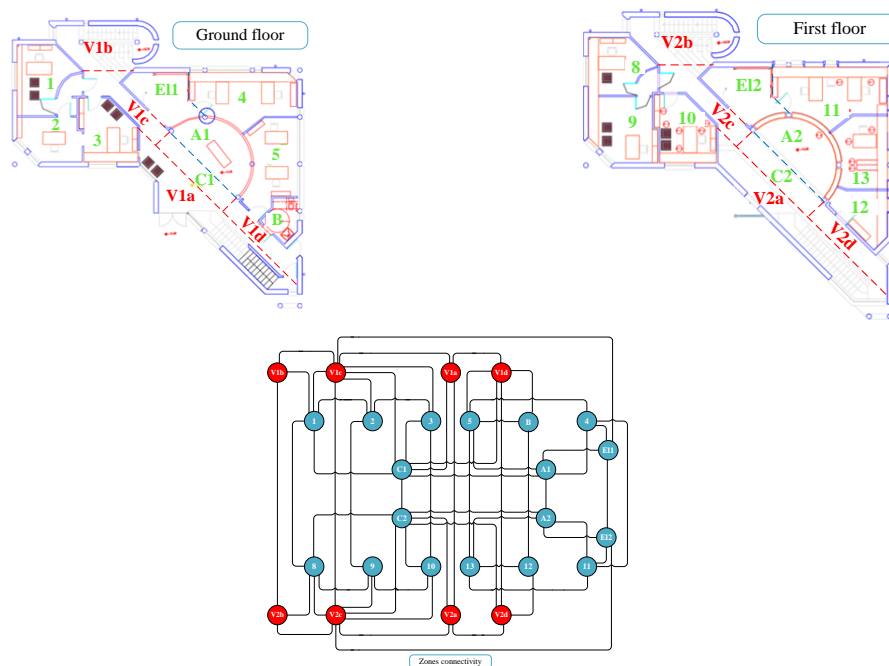


**Figure 4: Zoning of the TUC building (above); connectivity graph for the thermal interactions (below)**

<sup>6</sup> Certain simplifications are still made, especially with regard to radiative exchange in non-convex spaces, but this is beyond of the scope of the present document.

As mentioned above, the spatial discretization should follow the level of modelling detail. As an example consider below the spatial discretization of the TUC test-bed. Shown in Figure 4 (top) is the floor plan of the building and the spatial separation into thermal zones. The building has a triangular shape with two sets of offices connected by a long corridor that crosses the building.

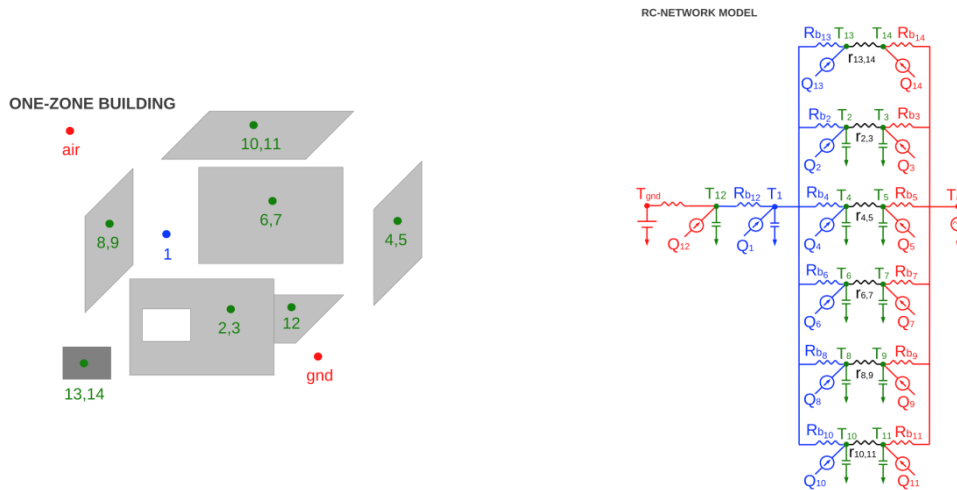
The presence of the solar atrium above zones A2 and C2, along with the radiation model used, gives an uneven heat distribution between the simulation and reality. For this reason, splitting the corridor into more zones, so that the solar gain and radiant exchange effects can be correctly captured is necessary. In Figure 5, a different zoning is introduced by splitting zones C1 and C2 in the original case to smaller zones. In the latter case, comparing simulation results can be shown to be in good agreement with sensor measurements obtained from a sensor placed on the curved wall of A1 [5], [6]. Compared to the quasi-static methodologies a more granular zoning is required here. Advantages of the additional modelling detail is that, at this level of detail, temperature and other variations can be correctly captured but at the expense of a more detailed and lengthy definition of boundary conditions, along with higher computational complexity. The zoning shown in Figure 5 is one of the many ways that the discretization can be defined. Many other choices for defining the zones exist, each of them yielding different model-level representations of the same building. As can be expected, assuming these different approaches are done in a way that all effects can be captured similar results should be expected: temperature histories calculated will be different, but such deviations will be “small.” This undefined degree of smallness, combined with the difficulty of having a uniquely defined zoning procedure, introduces an uncertainty that is hard to quantify. In most cases this does not pose a serious problem: the errors introduced due to “discretization” errors are of comparable or smaller magnitude, compared to the errors due to the zoning modelling assumption and uncertainties of other inputs.



**Figure 5: Zoning of the TUC building to account correctly for solar gains through the atrium (top); connectivity graph for inter-zone thermal interactions (bottom).**



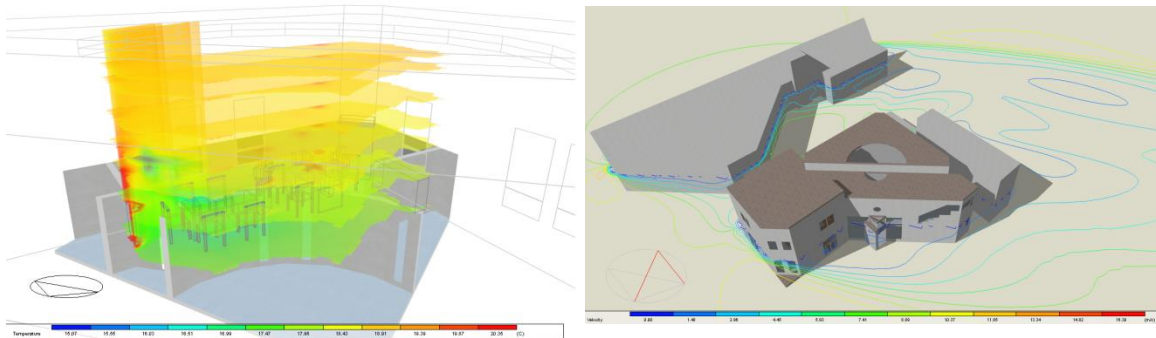
We mention another calculation methodology, of interest to BaaS, currently in development by the TUC research group: System of Resistances and Capacitances (SRC) [15].



**Figure 6: System of Resistances and Capacitances (SRC)**

SRC follows the zonal paradigm: the building is split into surfaces each of which comprises a number of surfaces. Information for setting up the spatial discretization is obtained from a gbXML data file. In a pre-processing step, the zones (nodes) and surfaces (links) define the connectivity graph. Each opaque surface element is modelled using a 1R2C representation (one thermal resistance, two thermal capacitances); representations for other opaque and transparent elements are similar. The SRC uses the connectivity graph to assemble the overall thermal network representing the building. In Figure 6, an example is given for a single zone: on the left a decomposition of the building to surfaces is shown; on the right, the generated thermal circuit is depicted. Heat sources are used to represent thermal interactions, e.g. due to solar gains. Within SRC, a detailed 3-D modelling is used to calculate radiative exchanges between internal and external surfaces; in that sense the level of modelling detail for SRC is similar to the one utilized by EnergyPlus and TRNSYS. Using the elemental assembly, a system of ODEs is formulated and then a numerical discretization scheme (e.g. Euler's method) is used to numerically solve the resulting conservation equations. One desirable design aspect of SRC is the use of the standardized gbXML data model, for data input. In that sense, a more direct link exists between CAD/BIM software (used to design the building) and the input to conduct the thermal simulation. The extension of SRC to utilize as input IFC data is currently under investigation – more information will be presented in future WP5 deliverables.

In the previous calculation methodologies only sensible heat calculations were described. It is also possible to include latent heat into the calculation methodology: this is particularly desirable in the presence of humidification and dehumidification systems. In these cases, the air of building spaces is considered a mixture of dry air and water in the vapour state [12]. The amount of water vapour present building spaces' air affect the temporal thermal heat storing capability of the air and the heat transfer rate between the air and neighbour building elements [13]. Therefore to specify the percentage of water vapour in building spaces' air, vapour transfers between the outside air or adjacent air volumes and the air volume of the space under consideration have to be accounted at every simulation time instant, by augmenting the system of energy conservation equations with mass conservation equations for the moisture content in the air.



**Figure 7: CFD calculations for the TUC test-bed: temperature distribution in a zone during heating (left); external CFD calculation to determine the pressure and velocity fields.**

The zonal approximation is acceptable for many envisaged and practical use scenarios, as it manages to strike a balance between accuracy and the errors introduced by uncertainties (values for thermal properties, occupancy and operation schedules, user actions, etc.). Still in some cases, looking at finer-than-zone scales can be justified. Such cases include: thermal comfort studies where the temperature distribution and its variations within a room or a zone must be known; or in ventilation studies, where the age of air can be an important parameter; or even for the determination of the placement of temperature sensors so that a good reading, representative of the average temperature in the zone, can be ensured. In all these cases, the granularity offered by considering zonally averaged parameters is just not enough. It is exactly for these cases, that the use of Computational Fluid Dynamics (CFD) calculation methodologies can come handy. Shown in Figure 7, are two examples, of the implementation of such methodologies for the TUC test-bed. Shown on the left, is the temperature distribution on one of the rooms during winter heating mode, to estimate the homogeneity of the temperature fields due to heating from the radiators. A second example is shown on Figure 7 (right), here an external CFD calculation was performed to find pressure and velocity fields, developed in the building due to the presence of winds and the interactions with nearby structures. The information shown, velocity field on a plane parallel to the ground, is especially useful for estimating the pressure coefficient on windows, so that better modelling of infiltration can be achieved.

The topic of CFD is very mature and has been developed, applied, and extensively validated in many fields where the dynamic behaviour of fluids (e.g. air) needs to be computed. In these methodologies, first the solution space is defined and appropriate initial and boundary conditions are defined. Then a space-filling partition is introduced using pyramidal or hexahedral elements. The size of these elements should be smaller than a characteristic length scale related to the size of the flow structures that have to be resolved. This partition is often defined in a conforming manner, where the computational grid is defined in a conformant to the boundaries fashion. Then the conservation laws are stated: typically the Navier-Stokes equations for mass and momentum conservation and the energy conservation equation. A discretization methodology, like the finite-volume or finite-element method is then used to discretize the partial differential equations, on volumes or elements defined in the partitioned space. As a result of the discretization a large system of equations, is solved numerically to yield approximations to the temperature, pressure and velocity fields.

The biggest problem for CFD implementation as whole building calculation methodologies is the need for boundary conditions, to be prescribed on all boundaries of the computation domain.

An approach which is often used is to first use a zonal-type approximation, which is seen as a cruder first step, to create the boundary conditions, and then use these boundary conditions to pose and solve the CFD problem. This can be problematic for complex geometries, as a data transformation process has to occur between different domains. As can be readily inferred the pollution due to uncertainties in the results of the zonal approximation, is propagated in the CFD calculations. Unless great care is taken in performing the transfer of information between the two approaches, the validity of the obtained results is always subject to scrutiny. In that sense, CFD for whole-building simulation can be helpful as a quasi-qualitative tool for understanding fundamental flow structures, but with little hope of matching real operation situations in great detail. It is for this reason, that the use of CFD calculation methodologies is mostly restricted in the design phases of a building lifecycle.

### 2.1.3 *State-space calculation methodologies*

Beyond the model-based approaches discussed in the previous Sections, models developed specifically for a particular purpose (e.g. control design or fault-detection) are often developed and utilized in their respective contexts. In this case, no “one-size-fits all” solution exists. The purpose of using these models is markedly different from the ones mentioned above: it is not for performance estimation, or foretelling the variation of relevant parameters, but rather their characteristics stem from their role in the context where they will be used. For example, for model-based control design purposes, typically state-space models adhering to certain mathematical constraints (e.g. linearity or quasi-linearity) are required. The accuracy of such models may not be of essence but rather their ability to correctly capture dynamics and sensitivities of the system that is being modelled, as this is the critical quality needed for control design. The accurate zonal models described above, are computationally expensive and in a form that is not amenable to control design. For computational efficiency, the number of states should be “small,” as repeated evaluations might be required within the control design context where they will be applied.

In the building application domain, the development of such models remains an open problem – although see [21] for a discussion and some preliminary suggestions. Towards defining such models many approaches are possible. In the case where historical data from building sensors are available, structured identification can be obtained [24]. These data can be used as an input to a multitude of identification methods [11], [22], and [21], so that data-driven model inference can be achieved. This situation is especially limiting as the data might not already exist, thus an online experiment to the real building has to be conducted. On the other hand, such an experiment can be quite expensive and the system might not be excited enough to capture all relevant dynamics [23], [8], and [7].

What seems more plausible is the use of a “detailed” (in the sense of the previous section) model, acting as a surrogate of the real building. Using structured identification, a simpler purpose-built model can thus be constructed [7], [21], and [8]. What is particularly attractive in this approach is that excitation necessary for the identification happens at the simulation level, so it is possible to excite the system in many ways that would be impractical, or even unrealistic, if they were to be applied in the real building.

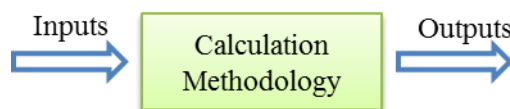
When computational effort constraints are present, model-reduction can be yet another approach towards a purpose-built simulation model. Here model-reduction [25] techniques are used to simplify (=reduce the dimensionality) of a given model, while retaining certain characteristics. Overall, model-reduction of non-linear models (as they appear in the context of building physics), can be a challenging task because one has to account for bifurcations in approximating

the solution manifold and the models obtained using such approaches, can only be valid locally in the solution space, with little guarantees with regard to more global approximation capabilities.

Within BaaS, such state-space models are both relevant and useful for the APO tasks. Providing support for such models, within the APO Kernel, is quite easy as they have fewer dependencies and simpler requirements compared to the transient calculation methodologies defined above. What is perhaps more interesting is how these models are developed and this largely depends on the service and algorithms that they support – this forms part of the modelling and algorithmic work in WP4 and WP5.

#### 2.1.4 *Uncertainties and Validation*

For all cases above, a calculation methodology can be seen as a well-defined deterministic calculation procedure, which encompasses models for the building and/or its constituents, along with means for establishing links, interactions and information exchange between the models. In all cases the modelling determines the type of inputs that have to be specified. The outputs are also determined from the nature of the calculation methodology.



**Figure 8: Calculation Methodologies – Input-Output Relationship**

The inputs are given as a set of objects, each forming part of a larger data model. A dictionary can be used to define the object names and the data model. A parser, using the dictionary, reads the input and processes them for syntactic correctness and completeness in the definitions. Irrespective of the data model used, defining the input comprises two parts:

1. Modelling definitions (e.g. selection of zoning, representation of the HVAC system), and;
2. Factual data and boundary conditions specification (e.g. occupancy profiles, material properties, weather data, etc.).

As can be expected, the accuracy of the predicted outputs strongly depends on the calculation methodology used, but also on the quality of the inputs. As discussed in the previous Sections the selection of zoning can have a profound effect on the quality of the modelling: too crude a zoning and it may not be able to correctly capture relevant physical phenomena; too fine a zoning and excessive effort might be required for setting up the problem, in addition to longer computation times. What is maybe comforting is that if (any) one of the many “valid” zoning approaches is selected, the value of the outputs will not strongly depend on the selection – although small variations in the output values should be expected. Missing or uncertain data for specification of the boundary conditions can be more problematic: erroneous inputs are propagated via the calculation methodology to affect the outputs.

In defining validation procedures for transient Calculation Methodologies all the issues above should be addressed.

The approach taken in:

- EN 15265: Thermal performance of buildings – Calculation of energy use for space heating and cooling – General criteria and validation procedures [2],

- EN ISO 13791: Calculation of Internal Temperatures of a Room in Summer without Mechanical Cooling – General Criteria and Validation Procedures [26]

addresses point 1 above by defining reference buildings that have only one zone. A specific set of boundary conditions and material and other parameters is prescribed regarding point 2. As part of the standard a set of test cases are defined, that must be passed by a calculation methodology to test compliance with the standard. This way any tool that is capable of accommodating the boundary conditions described in the standard can be tested for compliance.

Also a relevant standard for calculation methodology validation is the:

- ANSI/ASHRAE Standard 140-2001 Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs [27];

Here a set of synthetic benchmarks is defined (Cases) ordered in order of modelling complexity. The goal of this standard is to provide a standardized methodology for testing and debugging building energy analysis methodologies. As part of the standard, a well-defined testing procedure is established: if a calculation methodology fails a case, a number of diagnostic subcases are defined to help identify the root cause of the failure. Also, unlike the other standards, there are no fixed accuracy limits to determine that a calculation methodology has “passed” a case, rather a comparative approach is recommended, in which the results of the calculation methodology are compared against other state-of-the-art tools. This comparative methodology serves two purposes: first, to help diagnose modelling and coding errors; and second, to compare against other state-of-the-art approaches so that output variability due to different Modelling approaches can be better understood.

In many cases empirical validation studies, have been conducted against calculation methodologies, strengthening the confidence in the capabilities to correctly mirror reality. It should be emphasized that once a calculation methodology has passed all tests of a validation procedure, and is deemed “validated,” this in no cases does it imply that the calculation methodology represents the truth. It does show that a set of algorithms have shown, through a repeatable procedure, to perform according to the state-of-the-art.

Unfortunately, a simulation and its’ predictive capabilities are as good as the multitude of assumptions (regarding occupancy, plug loads, occupant behaviour, BEMS actions, use of “typical year” weather data etc.) that have to be performed for the input data. In the design and retrofitting phases, where these tools are typically used, reasonable assumptions regarding all aspects of building operation and equipment are made and consequently used in the simulation process. It is very often the case, that such assumptions prove to be wrong and, for this reason, real-world (measured) energy performance can vary significantly from the one estimated upon invocation of the energy simulation models.

## 2.2 Simulation Engines and Co-Simulation

A review of the previous sections reveals two fundamental simulation functional requirements: a) the availability of a mathematical model for describing all the components of the building, along with a proper solver for executing/simulating the model and providing outputs in each requested timestep, and b) the ability to simulate the model, using external inputs as values for certain model parameters (e.g. control schedules, occupancy patterns, etc.). Since for both the above requirements significant research effort is underway, a variety of software tools is available.

To start, for the definition of the complex mathematical model of the system, different flavours of detailed thermal simulation software exist. These tools provide a high-level, components-based interface, where building elements (like walls, materials, etc.) and systems (e.g. HVAC and RES) available in the software library are combined to formulate the model of the building. All these components are modelled using state-of-the-art approaches and mathematical definitions, are validated using a set of common test cases for detailed thermal simulation software, are constantly updated and some are free and open source. *Note here, that the use of a single software as it is the case of TRNSYS obey only to the possibility of integrating systems and building under one single solver and the accumulated experience of the work package 4 leader and some of the other WP4 collaborating groups with that simulation software that minimize the time needed to create new models, controls and simulation results.*

Moving forward, the necessity of combining models generated using heterogeneous modelling approaches or software has led to the definition of the Functional Mock-up Interface Standard (FMI), which is a standard to support model exchange and co-simulation of dynamic models. Here, each required model is compiled as a Functional Mock-up Unit (FMU) that provides a description on the input-output specification and execution requirements; subsequently all available FMUs are imported to the proper co-simulation environment and combined to formulate the simulation scenario. This approach allows, for example, to model the building envelope using software A, the HVAC system using software B and the RES using mathematical equations and combine all three components to a co-simulation setup, assimilating the behaviour of the system.

In the following sections, an enumeration of the high-level features of widely-used thermal simulation engines is provided, as well as an overview of the co-simulation hosting tool used within BaaS.

## 2.2.1 Whole Building Thermal Simulation Engines

### 2.2.1.1 EnergyPlus

EnergyPlus is software released by the U.S. Department of Energy. EnergyPlus follows the zonal thermal models paradigm, where the building is divided into spaces (thermal zones), each with a constant temperature, humidity etc. The energy conservation differential equation and the mass conservation differential equation on each zone are used to evaluate the evolution in time of the zonal thermal parameters.

In EnergyPlus structure the whole building system is divided into three main parts: Zone, System and Plant. The entire system consists of many interacting modules which are integrated and controlled by the Integrated Solution Manager. The schematic subroutine calling tree shows the overall structure of the program.

- ProcessInput (InputProcessor)
- ManageSimulation (SimulationManager)
  - ManageWeather (WeatherManager)
  - ManageHeatBalance (HeatBalanceManager)
    - ManageSurfaceHeatBalance(HeatBalanceSurfaceManager)
    - ManageAirHeatBalance (HeatBalanceAirManager)
      - CalcHeatBalanceAir (HeatBalanceAirManager)

The main input file is the input data file (IDF), an ASCII file which contains information about the building and the HVAC system to be simulated. The EnergyPlus input data are structured into classes. For each class, fields are defined, which describe the characteristics of the class objects. Objects are the instances of a class. All the available classes are listed into the Input Data Dictionary file (IDD).

The EnergyPlus Weather file (EPW) is an ASCII, csv format file containing the hourly or sub-hourly weather data needed by the simulation program.

### 2.2.1.2 *System of Resistances and Capacitances (SRC)*

System of Resistances and Capacitances (SRC) [15], is an element-based, hydro-thermal building simulation program which uses the electro-thermal analogy[32] representing a whole building as a System (or Network) of Resistances and Capacitances. The above approach enables the simulation output to be carried out as a solution of a state-space model (system of time dependent partial differential equations) over a finite time horizon. Essentially, SRC uses the building model data, to generate a system of combined heat and vapour mass transfer PDEs [14]. SRC performs simulations by solving numerically this system, in order to estimate temperatures, humidity ratios and atmospheric pressures developed inside building spaces. Additionally, using ideal heating and cooling devices of infinite capacity defined at every building space it estimates the energy and maximum thermal power demands required in order to retain the space temperatures inside a predefined range.

SRC uses as input model data gbxml files [37] and except from the main calculation routine, has also distinct sub-programs which can be executed separately and include:

1. A geometry program which essentially extracts geometrical definitions of building elements from the building model described by the gbxml file.
2. A material program which extracts the material properties of every building construction from the building model described by the gbxml file.
3. A view factors program which based on the geometry and material information extracted from the building model, calculates the view factors of building surface pairs [37]. View factors are used for long-wave radiation calculations [36].
4. A solar program which estimates the solar gains of building surfaces.
5. Schedulers, which are programs which define the operation schedules required as an input to the simulation process.

According to simulation calculation method classifications presented in section 2.1 the calculation method used by SRC can be classified as a zone-scaled, transient and hybrid type method. Consequently it is ideal for performing BaaS APO tasks.

Additionally, SRC uses a variable time step numerical scheme where the thermal input of every device (active: heating/cooling or passive: windows, blinds) can be altered in every simulation time step, according to the building conditions or by following a user defined schedule.

### 2.2.1.3 *Modelica*

Modelica is an object-oriented, equation based language enabling the modelling of complex physical systems, containing subcomponents, such as electrical, thermal, control, etc., using differential, algebraic and discrete equations. The large number of widely-available libraries modelling such subcomponents situates Modelica among the most powerful and popular simulation tools.

To start, usually a Modelica simulation environment is used, such as OpenModelica or Dymola, providing a component-based GUI for developing Modelica models, to speed up the design process. Consequently, the graphical model is converted to a corresponding text, containing the Modelica language equations describing the model. Finally, the code is translated to C and simulations can be performed.

For building simulation, the Modelica Library for Building Energy and Control Systems is freely-available, containing models for:

- Air-based HVAC systems
- Water-based heating systems
- Controls
- Heat transfer equations
- Multizone airflow.

A significant advantage of Modelica buildings library over other thermal simulation modelling tools, such as EnergyPlus is the way the warming-up process is performed. Here, a simulation is performed for a predefined warming-up period, using available measurements and the resulting building thermal state at the end of the warming-up period is saved in a text file. Subsequently, a new simulation is initiated and the saved thermal state is loaded and used as the initial conditions of the thermal simulation, thus reducing the computational complexity of tasks requiring frequent calls to the simulation model, like model-based control design optimization modules.

#### **2.2.1.4 TRNSYS**

TRNSYS simulation software (University of Wisconsin, Madison) is a transient systems simulation program with a modular structure where users design a determinate component order that represent real systems installed or to be installed. A wide type library covers most of the component possibilities able to be found in solar system facilities, HVAC systems.

The TRNSYS library includes many of the components commonly found in thermal and electrical energy systems, as well as component routines to handle input of weather data or other time-dependent forcing functions and output of simulation results. The modular nature of the software allows the possibility to create ad-hoc DLL models that represent particular problems not included in the software packages. The structure of a typical TRNSYS deck used in building simulation includes:

- Weather libraries in different formats as can be the Typical Meteorological Year (TMY) in its three different versions, EPW, German TRY's&VDI.
- Solar radiation models for fixed and tracked surfaces.
- Control components like PDI's, differential or iterative feedback controllers.
- HVAC components like Fan coils, Air Handling Units, Thermal collectors, boilers, heat pumps, pumps, valves.
- Electrical generators as PV collectors, Fuel cells, engines.
- Hydronic systems that connect the HVAC components.
- Building types. Especially type 56 represents in an exhaustive way the behaviour of zoned divided buildings.
- Connections to specific propose software as MATLAB (control), Fluent (CFD's), EES (Equation solver). Excel (Data treatment and Visual Basic Macros).



Each one of the previously named TRNSYS types has a FORTRAN code behind with the equations that define their principles and solve the equations that define their working modes. Some of the types need also of external files that define the efficiency machinery represented (i.e. Heat pumps need of a COP and Partial load files dependent of the two temperatures that define the heat sources)

Building types included in TRNSYS usually need of interface software (TRNBUILD) that makes easier the zone definition of the building, its control, set points, occupancy loads, scheduling and output settling to be provided as input for other TRNSYS types or to the program user, allowing him/her to get an accurate overview of the ongoing processes.

The simulation of buildings, understood as the structural envelope and its installed component required of at least three files:

- Simulation control where the solution method, equation solver, simulation times relaxation factors, constants, and tolerance values are defined.
- A file that defines univocally the building structure and the way it should be used.

Last TRNSYS version also includes:

- an extra file that defines an external shadow matrix that disturbs the solar radiation harvested buy the building envelope
- 3D building graphical definition importing IDF Sketch up files.
- Files that define the machinery efficiency under different *external* conditions.

### 2.2.2 Co-Simulation

Within Baas project, an enhanced (by the BaaS participants) flavour of Ptolemy II heterogeneous modelling and design framework, named Building Controls Virtual Test Bed (BCVTB), will be used as the co-simulation hosting platform. The BCVTB is a software environment, developed by Lawrence Berkeley National Laboratory which allows a dynamic data exchange between the two or more programs, at each time step of the co-simulation. The latest version of the software supports coupling of (building or system) models developed in Modelica, EnergyPlus, TRNSYS detailed thermal simulation engines, as well as models developed in a variety of programming languages, such as Fortran, C or Matlab. There is a preference in modelling the buildings using the TRNSYS for the reasons mentioned above, but through the co-simulation interface any other modelling tool (of the ones mentioned above) can be used. Due to this interchangeability (where the details are hidden behind the co-simulation interface), we use TRNSYS, SRC, or EnergyPlus in the component development with the understanding that all their differences are “hidden” behind the co-simulation interface.

In all cases the co-simulation data exchange between two programs is described by [31] and occurs as follows:

Suppose a system where two simulator clients are implemented and each simulator calculates a variables vector  $x_i$  which is strongly dependent on the variables vector of the other simulator at each simulation time step. In other words, simulator 1 calculates, for each time step  $k$ , the vector:  $x_1(k) = f_1(x_1(k-1); x_2(k-1))$  and simulator 2 calculates the vector  $x_2(k) = f_2(x_2(k-1); x_1(k-1))$ .

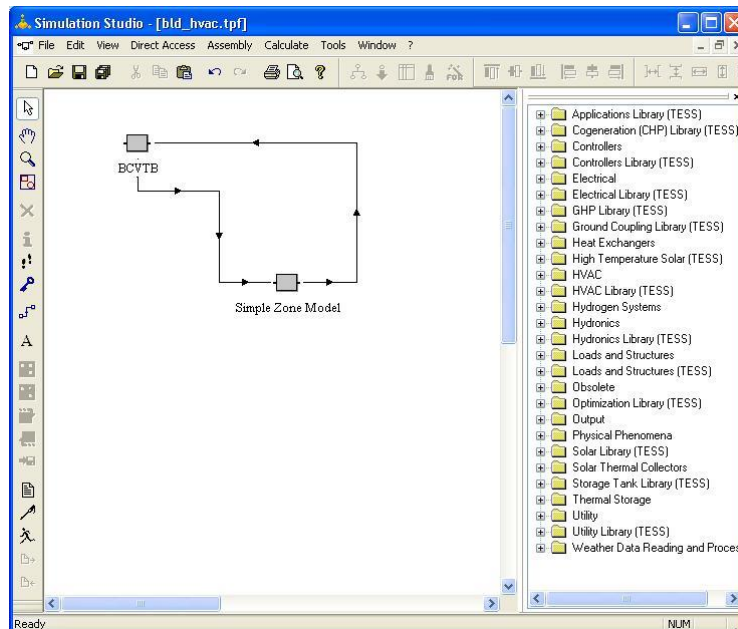
- For each simulator the initial conditions vector is defined  $x_1(0)$  and  $x_2(0)$ .
- At the end of the time step, simulator 1 sends the new variables vector  $x_1(k)$  to the BCVTB and receives the variables vector  $x_2(k)$  from the BCVTB. The same procedure is done with simulator 2.

- Simulator's synchronization (data exchange) is achieved in such a way that it does not matter which of the two simulators is called first.

In cases where, e.g. EnergyPlus constitutes the simulator 1 and Matlab constitutes the simulator 2, four steps are required to configure the data exchange:

- The essential objects of the variables we would like to exchange are also embedded to the input file of the thermal simulation model (IDF).
- An xml file that defines the mapping between EnergyPlus and BCVTB variables is created, named variables.cfg for the variables.cfg syntax).
- A Matlab script that defines the data exchange between MATLAB and the BCVTB variables is created.
- A Ptolemy model describing the data flow sequence is created.

Following, an example of a simple setup of heating control of two rooms in TRNSYS, *from the BCVTB website<sup>7</sup> is presented for illustration purposes*. Here, the model of the system is developed in TRNSYS and a type 6666 (BCVTB) component controls how the variables are communicated between the model of the building and BCVTB. In each time-step of the (co-) simulation, control inputs (possibly originating from external software) are passed to the zone model and the zone temperatures are communicated to the BCVTB environment.



**Figure 9: TRNSYS simple HVAC example**

<sup>7</sup> <http://simulationresearch.lbl.gov/bcvtb/releases/latest/doc/manual/tit-TRNSYSCon.xhtml>

### 3 Simulation task in BaaS and its functional requirements

As stated in the previous section, building simulation tools provide the opportunity to predict and evaluate the performance of a building, based on physically and mathematically methods. They allow the theoretical assessment of the complex building model, the simulation of its physical behaviour and the interaction among the components. Within the BaaS system simulation tools serve as main "P" task for the prediction of the buildings' behaviour regarding mainly the estimation of energy performance, and different user comfort parameters, but also providing values of expecting component/set of components efficiencies to be used in evaluation and fault detection analyses.

Thus, simulation modules within BaaS will provide prediction capabilities to any APO service that needs them. Basically, the referential (expected) behaviour (energy performance) of observed entity (zone, building, component, etc.) can be simulated. Or, for example, a test of proposed control strategy can be run against the relevant model and simulate thereby its effect.

Simulation is utilized to predict the building state under specific internal and external inputs/conditions. The only alternative is data-driven approaches that require significant amount of historical data in order to be able to predict a large variety of building states and even then exhibit poor extrapolation capabilities (i.e. the ability to predict future states from inputs not included in the training samples). This can be avoided with the use of "accurate enough" simulation models, where buildings and energetic systems installed in them can be accurately simulated, delivering as simulation outputs nearly real behaviours when different inputs are applied on them. This way, considering the model-based iterative optimization approaches utilized in WP5, the models developed in WP4 provide enough accuracy to evaluate different control strategies, applied with the same boundary conditions (i.e. weather conditions, occupancy schedules, equipment loads, etc.), necessitating only enough historical data for the model validation. The requirement to evaluate controllers with the same boundary conditions implies that the models used for the optimization tasks should be deterministic, which – in connection to the definitions of D5.1 – implies that the building and observation models are known and no noise is injected to the system. Of course, inaccurate weather forecasts and occupancy predictions add stochasticity to the process and necessitate frequent control re-designs, as described in D5.1.

The model includes the building envelope (geometry, material, shading devices, etc.) and the installed building services (heating, cooling, air handling, domestic hot water, lighting) as well as physical or mathematical representations of user behaviour and control plus boundary conditions (e.g. climate). The properties of the model heavily depend on the specific prediction task requirements. Thus, in case of the assessment of the annual energy demand for heating, a single-zone model with a large time-step and an approximated reduced model for the heating system, without necessitating details on the operation of the building (e.g. shading devices), would suffice. On the other hand the study of thermal comfort (e.g. to estimate the periods of overheating in certain rooms of the buildings) requires a detailed and accurate model of equipment, usage and control with adequate simulation time steps of a few minutes and should take into account transient (dynamic) processes due to the building mass.

From the discussion above, it is obvious that the simulation/prediction tasks within BaaS differentiate from common – usually year-round whole-building – simulation setups, in the following:

- Simulation period can be quite short, varying from few hours to few weeks, depending on the task;

- A simulation process can require data from different time-periods simultaneously, like e.g. simulations for control design optimization processes. Thus, the model might need:
  - Historical, measured data for the warming-up process, allowing to assimilate the initial thermal conditions of the real building;
  - Real-time data, provided by the BMS;
  - Predicted data, for the weather and occupancy conditions, as well as equipment loads.
- The simulation model can include only specific subsystems of the building (and sometimes even neglect the envelope) for specific tasks;
- The simulation requires inputs from external sources (i.e. weather predictions, historical data, control decisions), thus necessitating the setup and utilization of a co-simulation module.

In this sense, simulation models are consumed by APO services that provide useful functionalities with respect to the building operation. Such APO services can address various building optimization opportunities and therefore are of different complexity, but many of them rely on various simulation models. This is especially true for APOs in Automation and Management system layers (see D5.1). Generally speaking, all three types of APO services groups require some model:

- **Fault Detection & Diagnosis (FDD)**, where services for comparing measured key performance indicators (KPIs) to simulated ones are defined;
- **Energy Management**, containing services that perform baselining, where the referential energy consumption is constructed from a validated model of the building;
- **Control Design & Optimization**, where model-based control design approaches (e.g. MPC or Reinforcement Learning) are defined.

Table 3 summarizes the necessity for a *model of the process* at study (e.g. HVAC components, energy systems, buildings, neighbourhoods, etc.), while WP4 is responsible for providing proper models, suitable for each task.

**Table 3: APO services model requirements**

APO services type	APO Modules	Model
<b>FDD</b>	Knowledge-based equipment/device fault discovery	
	Model-assisted equipment / device fault discovery	X
<b>Controller</b>	Knowledge-based	
	Function Representation	X
<b>Control Design Optimization</b>	Data-driven	
	Model-assisted	X
<b>Energy Management</b>	KPI calculation	X
	Baselining/benchmarking	X

---

In detail, simulations will be used with BaaS to accomplish a variety of different tasks:

- **Energy performance estimation.**

In this task the energy performance of the whole building is estimated. Energy performance includes total energy needs, including energy used for conditioning the spaces. In the transient calculation methodologies above, thermal comfort parameters can also be computed.

- **Energy performance forecasting.**

The goal of energy performance forecasting is to estimate building energy needs in order for comfort conditions to be preserved in building spaces, during a finite future time horizon. The use of forecast data obtained from various sources is necessary in this case. As can be expected, the validity of the forecasting process depends strongly on the quality of the forecasts. Integration of past data and forecast data (obtained from different sources) is essential here and the abstractions of the middleware will facilitate access to these data, so that the problem can be correctly set up.

- **Model calibration.**

Although models are designed to predict the real behaviour of buildings and their systems as accurately as possible, their predictions may differ from real sensor measurements, because of a different number of reasons including modelling insufficiencies, or incorrect model parameter values estimation. Model calibration tasks rely on past sensor measurements in order to change the model parameter values and bridge the above gap.

- **Virtual sensors.**

In many cases there are building locations where sensors cannot be placed or they are not present. In such occasions, the missing sensor data can be replaced by simulation value estimates by performing a task called “Virtual sensors”. Assuming a well conducted simulation, the computed values – “virtual” sensors – can yield acceptable approximation of real parameters.

- **Components validation.**

System performance can degrade over time, leading to out-of specification operation. This can have adverse effects with respect to energy performance and thermal comfort. Anomaly detection and identification using simulation-based methodologies can be one of the ways, to identify such events.

- **Control design.**

The general purpose of (supervisory) control design is to design controller that given state parameter values will return operation schedules and commands of controllable building elements. In model-based control design the calculation methodology (here synonymous to “model”) is used in combination with model-assisted control design optimization algorithms to generate such strategies.

- **Control design optimization.**

The generated control actions, using simplified state-space models, can have poor performance when applied to the real system. For this reason, the resulting controllers can be improved using more “accurate”<sup>8</sup> building models, by performing a second optimization step. Uses of calculation methodologies can be an invaluable asset in fine-tuning/optimizing controller parameters.

In order to determine suitable models for each task, as well as ensure a high level of accuracy, the following functional requirements have to be determined:

- **Model type:** depending on the specific APO service properties, different model types may be applicable; especially for detailed thermal simulation models, an analysis on the respective calculation methodologies<sup>9</sup> is in order;
- **Input Data:** for accurate modelling, two categories of input data are required: static data, containing information on the geometry, construction and actuating and energy systems; and dynamic data, including factual (sensed) data, along with forecasts for pertinent parameters (e.g. weather, occupancy) used to bridge the “simulated” and “real” worlds, reducing or even mitigating design-phase uncertainties.
- **APO Layer integration:** all the defined simulation tasks have to be configured and made available to the APO services through a properly defined interface, allowing transparent invocation of all different simulations.

### 3.1 Model type requirements

#### 3.1.1 Calculation Methodologies to be supported in BaaS

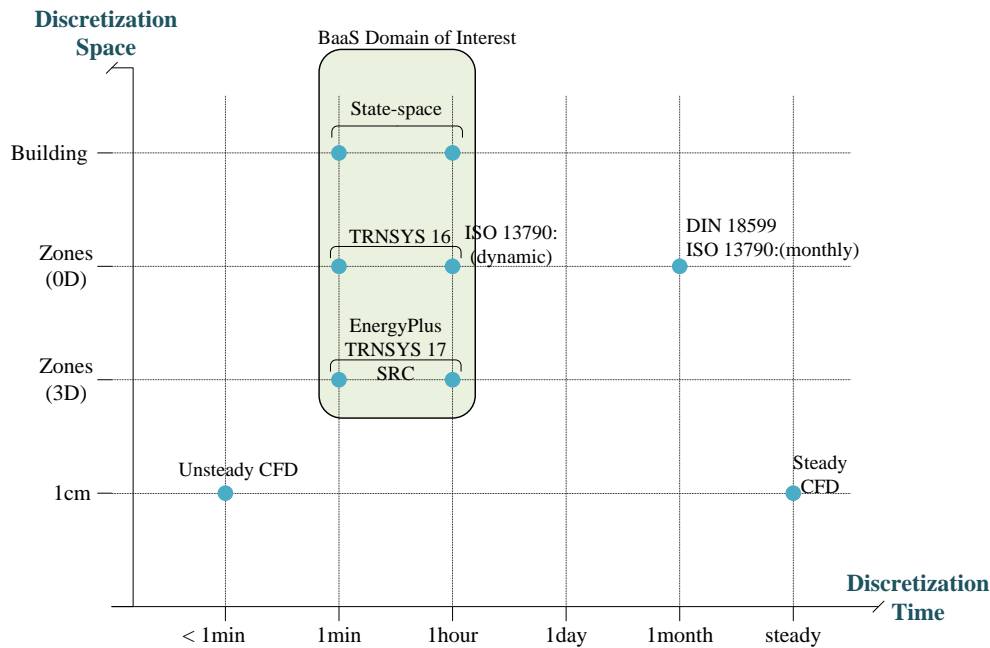
In BaaS we intend to provide a more active role of simulation and such calculation methodologies in the building operational phase. These calculation methodologies are to be registered within the BaaS framework, made available through the APO Services Kernel and the simulation module component, to be consumed by APO Services – see D5.1 for information on which APO Services need to access simulation components. As mentioned in Section 2 quasi-static and CFD calculation methodologies are primarily useful in the design phases, either due to the resolution of their predictions (annual basis for quasi-static) or due to the inherent assumptions and modelling detail required – as such, they are of lesser importance within BaaS. The use of time-steps in the range of a minute to one hour allows to account for the dynamics of active climate control systems, but also to incorporate control strategies that use state measurements as inputs to compute actuation commands. The desire to use simulation as a forecasting tool, also suggests that a “small” time step might be warranted. In view of the

---

<sup>8</sup> More accurate than state-space models.

<sup>9</sup> A note regarding terminology is in place: we use the term *calculation methodology* to describe a clearly-defined procedure (however complex it might be) toward performing a calculation of interest (e.g. energy performance estimation). Oftentimes mathematical *models* are used to represent components of the system in consideration; the use of such models can form part of this calculation methodology. *Simulation* refers to the implementation and execution of the calculation methodology. In this text, the terms *calculation methodology*, *simulation*, and *simulation models*, are used synonymously, understanding the (admittedly small) risk of ambiguity. Thus, the term *simulation module* refers to the software component implementing the particular calculation methodology.

comments above, in **¡Error! No se encuentra el origen de la referencia.**, the type of calculation methodologies of interest to BaaS can be identified.



**Figure 10: Amalgamation of discussion in this section**

Shown in **¡Error! No se encuentra el origen de la referencia.** are the calculation methodologies discussed in the previous Sections, on a diagram with the spatial and temporal discretization on its two axes. The classification based on the spatial discretization is important as it determines the level of modelling detail and the amount of information that has to be prescribed as input, when defining the geometry and other related information. The temporal discretization dimension is also important as it determines the integration time step, and consequently the granularity in which dynamically changing data (occupancy, weather, etc.) should be defined.

In conformance with terminology used in EnergyPlus we will call *Schedule* any time-varying function<sup>10</sup> (occupancy schedules, window opening schedules, etc.). In general input data can be classified with respect to the frequency of their change into:

- Static Data (SD). SD include: the building geometry, construction materials, glazing information, systems used in the building, etc.;

<sup>10</sup> The term *Schedule* adopted here should not be confused with the definition of Schedules in other contexts. In EnergyPlus the use of the term originates from the Schedules defining operation of components (Heating season schedule, Cooling season schedule) or to describe occupancy (Working Schedule). Used as design tool, EnergyPlus needs these parameters to define operation of the HVAC systems, or to compute heat gains due to occupancy. In reality, occupancy could be measured using an occupancy sensor, and the detailed “Schedule” (sensor measurement) could be introduced into the simulation. To maintain consistency in terminology, while accommodating for all such cases, we define *Schedule* to be any time-dependent function that might need to be provided to the simulation engine.

- Dynamic Data (DD). DD consist of all time-dependent data: user-actions (e.g. opening and closing the windows), occupancy schedules in each of the building zones, use of equipment, weather predictions, BEMS commands, etc.

Static data are relatively well-known and can be introduced in the simulation. As stated above, uncertainty mitigation in the inputs is important for validated calculation methodologies, to ensure good agreement between simulation and reality. In the design phase it is often hard to define with good accuracy the dynamic portion of the data, and reasonable assumptions are made – quite often these assumptions are invalidated in the operational phase. During operation there are two types of information sources that we can exploit: an IFC-based BIM has all the data structures to store and describe static building information; and, a Data Warehouse, where dynamic data obtained from sensor measurements inside the building can be utilized to comprehensively partially cover the “uncertain” dynamic portion of the input data. Both of these sources of information are addressed within BaaS through the definition of the extended BIM (eBIM), described in more detail in WP2 deliverables. Other sources for obtaining dynamic data can include external services: weather forecast service, room scheduling services, etc. A further explanation about input data is given in the following section.

### 3.1.2 Simulation models

In view of the prediction tasks required for the APO services, the architecture of the APO kernel (with respect to model use) emerges. Here, since the APO services “consume” *process models* for performing their assigned tasks, the APO kernel should be able to host and invoke various types of process models (e.g. state-space models, detailed thermal simulation models, etc.), without providing support on the implementation of these models. In other words, the *simulation models of the building components* (like PVs, HVAC systems, etc.) should not be “hardcoded” to the kernel, but the kernel should be able to invoke process models from the available models library.

Models can be invoked in different ways and for different reasons from WP5, delivering to the APO services (WP5) the demanded results that lately will be worked inside each APO function. As an example, it can be created simulation models for:

- **Components:** where their efficiencies can be compared to the real measured ones searching for inefficiencies, failures or wrong control modes applied. For example, a boiler working out of its temperature ranges has lower efficiency and potential failures risk. Or for the same boiler can be seen how unexpected higher return temperatures decreases the delivered power;
- **Set of components:** where the interconnection of the elements creates different dynamics depending on the control applied. For example, the energy delivered by a radiator to an occupied space depends highly on the thermal level of the working fluid due to the radiative/convective design of the element. Lower sending temperatures due to the inertial behavior of intermediate junctions or tanks derive in less energy delivered to the controlled zones and high discomfort periods, so the test of different control strategies allow to maximize the energy utilization;
- **A building zone or group of zones:** where the occupied spaces must have minimal comfort conditions to be developed a determinate activity by the people that work in them. For example, in order to minimize the energy consumption of the building keeping the comfort conditions controlled, some zonal models can be invoked with different activation temperatures on the active elements, or control strategies to be applied on the zones that save energy along the studied period;



- **Combination of building and components:** where the set energy produced/energy demand is optimized to keep them as close as possible, minimizing energy losses. For example, the integration of renewable energies on the building allows consuming the maximum of that “free energy” produce, minimizing the external dependency of the building.

Note that models belonging to the above types (or combinations) may be invoked multiple times by the APO services.

Moving forward, in order to define a prediction task, two components are required: the static part of the model and dynamic schedules (historical data, predictions, etc.) to assimilate the actual thermal state of the building under proper boundary conditions.

### 3.1.2.1 *Static part of the model*

Even though model-based BEMS design approaches can yield significant energy savings, the model development cost hinders wide market adoptability. For this reason, within BaaS a semi-automatic model generation process has been envisioned and is under development (WP4). Here, as a first step, static information on the building, such as geometry, materials and system properties are retrieved from the available BIM repository and the information is processed and translated to the model. More precisely, the following data are required:

**Geometrical Data:** Building geometrical data refers to all the geometrical information that is needed to describe the building; this includes all architectural elements. During the building design phase the architectural view is developed using CAD software. From the simulation viewpoint, the amount of information provided is too detailed for thermal simulation purposes. While the building elements such as interior/exterior walls, floors, roofs and openings assume to be monolithic in the architectural view, their subdivision into thermal boundary surfaces is a prerequisite for energy simulation purposes. Hence, reduction, simplification and transformation of the data provided by the architectural view are the thermal simulation geometrical requirement.

With respect to the transformation, the process of subdividing architecturally defined surfaces into thermal boundary surfaces is a difficult geometric operation and it is more reliable to be performed by the CAD tool being used to during the building's design phase. This way, a space boundary data object, part of the IFC data model, differentiates the way that space boundaries-surfaces are defined for architectural and thermal simulation purposes as follows:

- 1<sup>st</sup>-level space boundaries - the building's spaces are defined by space boundaries according to the architectural view;
- 2<sup>nd</sup>-level space boundaries - the building's spaces are defined by space boundaries according to the thermal simulation view.

**Building materials:** Every building element is associated with a construction, which describes a layer bedding where each layer corresponds to a one or more materials. Accurate definition of materials of a construction is a prerequisite for developing a simulation model that assimilates the thermal behaviour of the real building. For example, some materials associated with glazing surfaces (windows, glass doors, etc.) permit a fraction of sunlight to pass through to building interiors. This fraction of sunlight is represented by the solar heat gain and solar transmittance coefficients, which should be provided for every glazing construction and for different values of incident solar beam angles. In addition, all the construction materials impede and store thermal energy and permit by a fraction or completely block the sunlight. Therefore properties associated with these functions are universal properties as the knowledge of their values is required for all the materials.

All material properties are also described to the building model stored to the BIM repository; thus within BaaS each building element is associated to the respective construction, using this information.

### 3.1.2.2 *Dynamic part of the model*

As described in the methodological approach, models are used to evaluate the performance of various controllers. This means that the models are required to *predict* the behaviour of the building and the energy production systems. In order for this prediction to be accurate, there are two prerequisites: the static information of the building properties, as well as the forecasts on future disturbances (weather and in-building conditions) is accurate.

In addition, an important and rather overlooked parameter affecting the accuracy of a model is determining the initial state of the system. Here, although the most accurate way to assimilate the initial state is to provide sensor information on the building state to the model, a significant amount of this information cannot be acquired, since it requires expensive and specialized sensors (e.g. measuring the internal wall temperatures in a concrete building). Thus, the fall-back strategy is to use historical data from the actual building information (zone temperatures, control actions, etc.) as inputs to the model and simulate a number of previous days. This process is commonly referred to as “*warming up*” process.

In more detail, the following dynamic data categories are required by the model and affect its accuracy:

- **Weather Conditions:** Weather conditions significantly affect the behaviour of the building and are one of the major disturbances to the system, thus accurate weather forecasts are essential for enhanced model accuracy;
- **Occupancy schedules:** Generally, occupancy is a parameter which is difficult to estimate. In order to predict future occupancy values statistical data obtained by the data warehouse can be used, along with specific information, such as vacations, leaves of absence, etc. Consequently, past and predicted occupancy data are injected to the model;
- **Internal gains schedules:** Operating building equipment (computers and electrical equipment) as well as the presence of people act as internal thermal sources (air and surrounding internal surfaces). Since this total amount of thermal energy is a general nonnegative number varying with time, it can be represented by a schedule (a time varying nonnegative continuous function). This schedule is called internal gains schedule and is assigned to every building space. Usually the internal gains are estimated based on the number of people being inside a building space and the operational schedules for equipment in the same space;
- **Effects of User Behaviour:** Apart from weather conditions, the second significant uncertainty affecting the accuracy of the model is the disturbances coming from user behaviour. Here, we don't refer to alterations on the occupancy schedule, but the user actions upon the uncontrollable devices of the building (e.g. opening and closing windows). Creating forecasting models for such user-behaviour is nearly impossible and this is one of the reasons the controller is re-designed in pre-defined time-intervals; this way, the disturbances are injected to the model through the warming-up process and the controller to be applied to the building will be able to react to them;
- **Controllable device schedules:** Devices controlled by Baas system belong to the class of controllable devices. While past actions of such devices are logged to the system and are passed to the model, future control actions are generated using the controller in each timestep of the simulation.

When all data are available, different controllers are evaluated to the model by maintaining all other dynamic data the same, while injecting different controllable device schedules (controllable inputs), designed by each controller tested. Finally, a set of KPIs outputted by the model can perform the optimization tasks. These model outputs are the variables appearing in the cost function to be minimized (e.g. thermal energy consumption) and the constraints defined for such optimization (e.g. comfort conditions).

It is obvious that a controller evaluation is based on the ability to perform simulations using dynamic schedules for the data categories described above, while interacting dynamically with the control software. While this might be trivial for the case of analytical models (e.g. state-space models), for detailed thermal models is achieved using the *co-simulation* method which consists of the use of different software components for run-time coupling.

### 3.2 Input Data Requirements

In the previous Section, static and dynamic data needs have been described. In this Section, we take a deeper look into input data, in an effort of classification based on the type of information which is required. The discussion in this Section, helps make the types of data needed more concrete along with outlining some problems that have to be addressed within WP4 and WP5.

- **Building Model Data (BMD).** BMD include static information about the building such as the geometry of the building components (walls, floors, and openings), their material properties and the characteristics of the systems installed in the building spaces.
- **Schedules (SCH).** Schedules encode all dynamic data.
- **Weather data (WTH).** WTH data refer to parameters characterizing the building prevailing conditions which consist of the air surrounding the building and the ground space it is founded on.
- **Other Simulation Parameters (OSP).** OSP include data referring to the simulation time limits (starting and ending time instances) and the types of the requested simulation outputs.

As was mentioned in a calculation methodology multiple outputs might be returned. It is part of the parameterization and initialization of the Simulation objects in BaaS to define which the inputs of interest are. This information is included in the topic OSP defined above.

#### 3.2.1 Static data

##### A. Building geometrical data

Building geometrical data refers to all the geometrical information that is needed to describe the building; this includes all architectural elements. During the building design phase the architectural view is developed using CAD software. From the simulation viewpoint, the amount of information provided is too detailed for thermal simulation purposes. While the building elements such as interior/exterior walls, floors, roofs and openings assume to be monolithic in the architectural view, their subdivision into thermal boundary surfaces is a prerequisite for energy simulation purposes. Hence, reduction, simplification and transformation of the data provided by the architectural view are the thermal simulation geometrical requirements [20].

With respect to the transformation, the process of subdividing architecturally defined surfaces into thermal boundary surfaces is a difficult geometric operation and it is more reliable to be performed by the CAD tool being used to during the building's design phase [19]. This way, according to [10], a space boundary data object, part of the IFC data model and named

IfcRelSpaceBoundary, differentiates the way that space boundaries-surfaces are defined for architectural and thermal simulation purposes as follows:

- 1<sup>st</sup>-level space boundaries - the building's spaces are defined by space boundaries according to the architectural view
- 2<sup>nd</sup>-level space boundaries - the building's spaces are defined by space boundaries according to the thermal simulation view

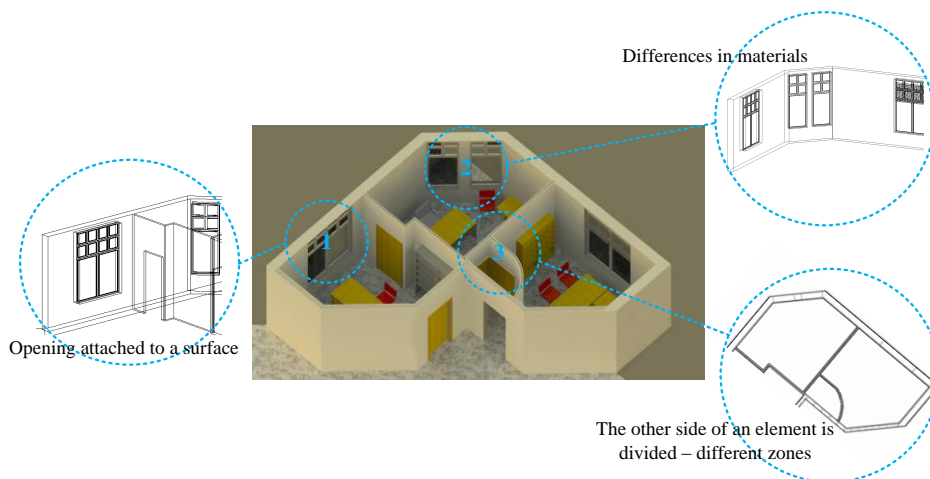
1<sup>st</sup>-level space boundaries are the boundaries of a space defined by the surfaces of building elements bounding this space. These space boundaries do not take into account any change of material in the bounding building elements, or different spaces behind a wall or slab. 1<sup>st</sup>-level space boundaries form a closed shell around the space and include overlapping boundaries representing openings in the building elements. In order to describe voids, there are 1<sup>st</sup>-level space boundaries with holes and separate 1<sup>st</sup>-level space boundaries representing openings which overlap and are coplanar with the space boundaries representing the host space boundary.

On the other hand, 2<sup>nd</sup>-level space boundaries still represent building elements that bound the space, but these elements are subdivided in cases where:

1. openings are attached to a space boundary,
2. differences in materials of a space boundary occur, and
3. the other side of a building element is subdivided due to the existence of a wall separating different spaces.

2<sup>nd</sup>-level space boundaries represent both sides of a heat transfer surface separated by the thickness of the building element. The building geometry description by 2<sup>nd</sup>-level space boundaries is a prerequisite for the thermal simulation purposes. They can be used by thermal simulation software, but the combination of the two adjacent surfaces to form a single heat transfer surface is required.

The connection geometry of 2<sup>nd</sup>-level space boundaries is restricted to planar surfaces only. Curved surfaces must always be segmented since, 2<sup>nd</sup> level space boundaries are described by planar surfaces only. Overlapping between 2<sup>nd</sup> level space boundaries of building elements and their hosted openings is not allowed. Hence, in order to describe a void in a wall, two different common orthogonal surfaces representing the host element and the opening are defined.



**Figure 11: Cases where building elements are subdivided – TUC building example**

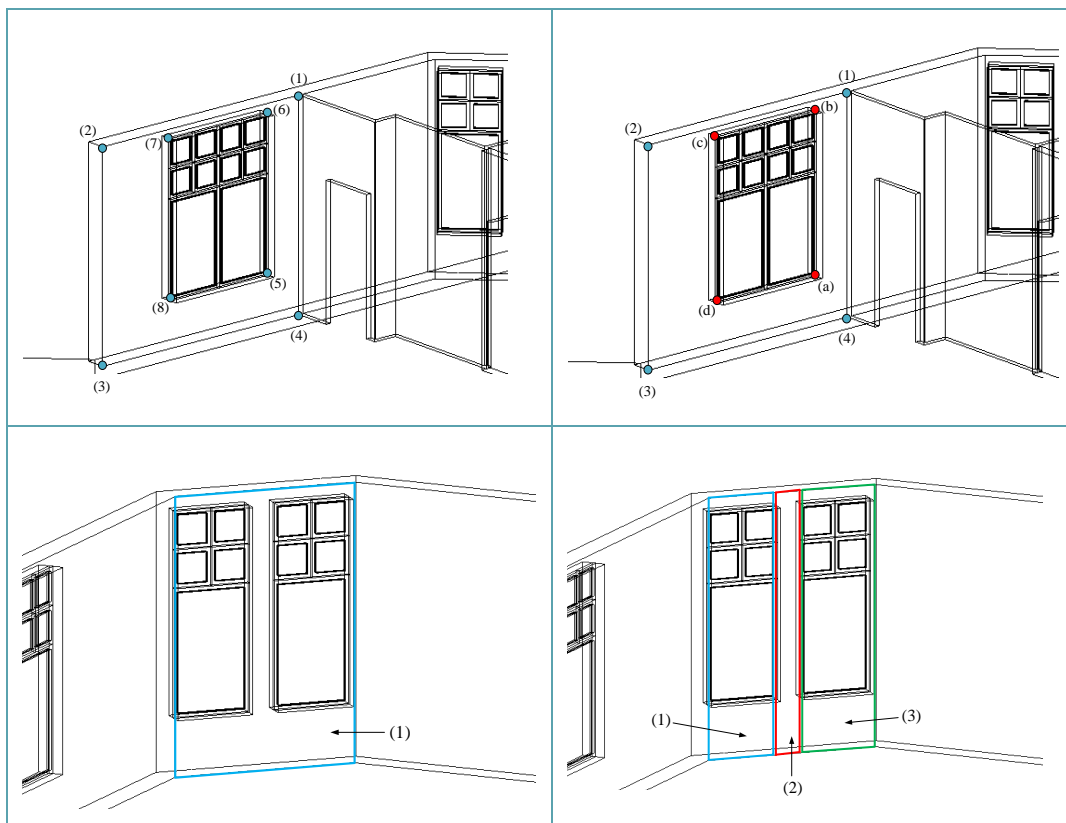
Figure 11 is a part of the three-dimensional geometry of TUC building, developed in Revit 2013. In this figure three examples are identified where subdivision of building elements is required (the numbers 1, 2 and 3 correspond to the order in which the cases are enumerated above). Table 4 presents the differences between 1<sup>st</sup>- and 2<sup>nd</sup>-level space boundaries for these three examples.

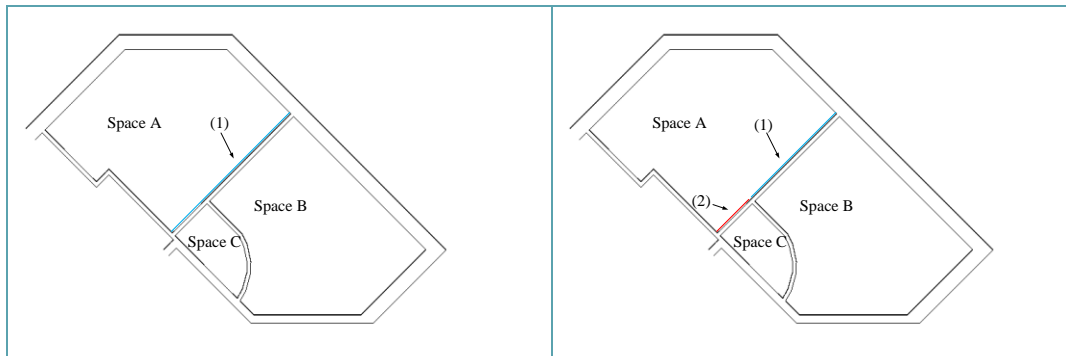
As Figure 11 depicts, in case 1 an opening is hosted to a wall. In 1<sup>st</sup>-level space boundaries description, a wall with a hole, described by 8 points, is defined (see the 1<sup>st</sup> row of Table 4); 4 points describe the vertices of the wall and 4 the vertices of the hole while at the same time, 4 points are required for the description of the opening. In 2<sup>nd</sup>-level space boundaries description, only 4 points are required for the wall definition. The opening constitutes a different space boundary and is defined separately from the wall.

Assuming that the construction of the gap between the two groups of windows differs from the construction of the rest of the wall in case 2, differences in materials of a wall occur (see the 2<sup>nd</sup> row of Table 4). For the definition of this wall at 1<sup>st</sup>-level, the description of one surface is sufficient; while at 2<sup>nd</sup>-level three different surfaces need to be described.

In case 3, the other side of a Space A boundary is subdivided due to the existence of a wall separating space B and Space C (see the 3<sup>rd</sup> row of Table 4). Here, at 1<sup>st</sup> level space boundary description, this Space A boundary is described by one surface while at 2<sup>nd</sup> level, its division into two surfaces is required.

**Table 4: Describing the differences between 1<sup>st</sup> (left) and 2<sup>nd</sup> (right) level space boundaries – TUC building example**

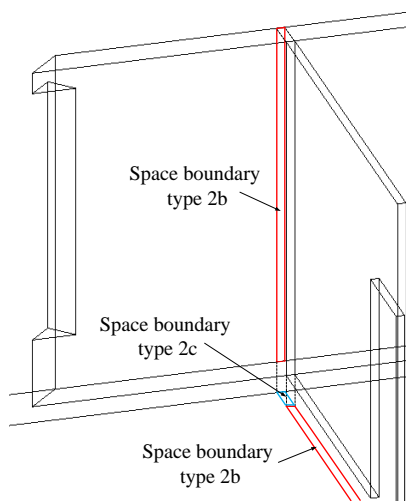




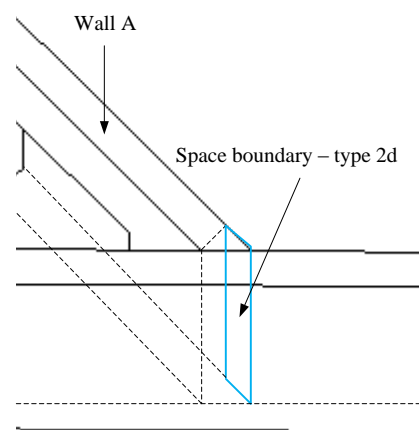
Regarding the data reduction and simplification, there are some special types of 2<sup>nd</sup> level space boundaries and are observed when the element behind the boundary is a building element. Figure 12 and Figure 13 present three common cases of special types of 2<sup>nd</sup> level boundaries.

Figure 12 shows two surfaces created by the projection of a wall element on its attached surfaces (red orthogonal) and a surface created by overlapping of two building elements (cyan orthogonal). The definition of these two special types of 2<sup>nd</sup> level space boundaries is required only in cases where these projections constitutes surface of another space and simulation requires complete enclosure of zone volumes.

In Figure 13, the two sides of wall A have different lengths due to the angled wall connection. The extra length on one side (cyan orthogonal) is defined to be the third special type of 2<sup>nd</sup>-level space boundary and it is not a geometry data requirement.



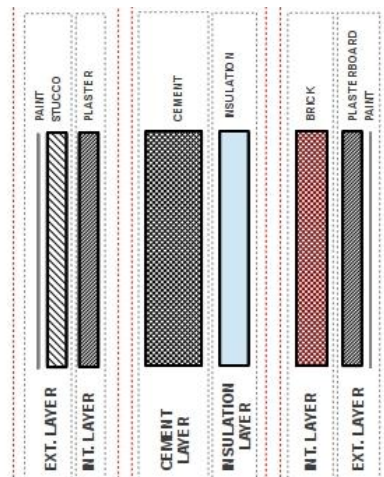
**Figure 12: Projections of a building element and overlapping of two building elements**



**Figure 13: Angled wall connection**

## B. Building material data

Every building element (2<sup>nd</sup> level spaces boundaries excluding its special types) is associated with a construction. A construction describes one layer bedding where, each layer corresponds to a one or more materials as the example of Figure 14 displays.



**Figure 14: Example of construction of building elements**

Every material present in the building's construction must be listed from the inner to the outer by convention. For example in an external wall the material attached to the inner space air should be listed first and the material attached to the outside air should be listed last. Different materials and their properties, play different roles during the calculations performed by simulation calls.

Opaque materials of a construction absorb shortwave (solar) radiation and emit and absorb long-wave (infra-red) radiation. Consequently, knowledge of the values of their properties, referring to these phenomena is required only for the external and internal materials. These properties are characterized as Opaque properties, as their values are required only for the external and internal materials of the constructions.

Furthermore, some materials associated with glazing surfaces (windows, glass doors, etc.) permit a fraction of sunlight to pass through to building interiors. This fraction of sunlight is represented by the solar heat gain and solar transmittance coefficients, which should be provided for every glazing construction and for different values of incident solar beam angles. Material properties associated with only glazing constructions are defined as glazing properties.

Finally all the construction materials impede and store thermal energy and permit by a fraction or completely block the sunlight. Therefore properties associated with these functions are universal properties as the knowledge of their values is required for all the materials.

Building material properties are obtained using BIM data queried from the IFC file. Some of the data could also be provided by the data warehouse if the respective elements refer to the building envelope. Phase change materials and green roof materials alter their thermal properties as their state change and require more data to be defined.

### C. Building Systems and Components

Building systems refer to a variety of devices ranging from active (energy consuming) micro-climate control devices (fans, HVACs, heaters) to passive devices (consuming zero or considerably smaller amounts of energy than active) such as mechanical blinds. Generally systems include any device which might be uninstalled or installed during the building's operation and affects the conditions of building spaces. Knowledge of the way these devices affect simulation parameters is of paramount importance. Building systems can be classified into the following categories:

### 1. Daylight control systems (DCS)

These types of systems affect the luminance of building spaces. They control the artificial lighting equipment of building spaces depending on whether the luminance levels measured by light sensors are acceptable.

### 2. Shading Control Systems (SCS)

These systems control the luminance (directly) as well as the thermal (indirectly) parameters of building spaces by completely or partially blocking the sunlight passing through building openings.

### 3. Airflow Control Systems (ACS)

These systems control the air quantity entering to or leaving from building spaces. They include natural ventilation devices such as mechanical openings and also forced ventilation devices such as fans.

### 4. Thermal Control Systems (TCS)

These systems affect mainly the temperature of building spaces. They include active devices such as heaters and cooling devices.

### 5. Humidity Control Systems (HCS)

HCS affect the percentage of water vapour present in the air of building spaces. They include devices which either increase this percentage (humidifiers) or decrease it (dehumidifiers).

### 6. Energy generation systems (EGS)

These are systems which are capable of producing energy making use of natural resources such as sunlight (photovoltaic panels) or wind (wind turbines).

Apart from the above classification they are building devices which belong to more than one of the above categories. The definition of systems belongs to the static category, although data related to their operation, and interaction with other building elements is defined as dynamic and appears in the Section below.

## 3.2.2 Dynamic data

### A. Schedules

Simulation programs require timing signals referring to the operation of devices including energy consuming, climate control equipment (such as HVACs, heaters, coolers) and passive devices (such as openings and blinds). These timing signals are in a broad sense, time dependent continuous functions which determine the operation state (on/off) as well as the operation mode characteristics of these devices. These functions belong to the general category of schedules (SCH). To define the values of the schedules used by the simulation programs, the time functions they refer to, have to be sampled at the simulation time instances.

#### A.1 Occupancy schedules

Building spaces often remain unoccupied during specific time periods. By turning off energy consuming devices, during these unoccupied periods, substantial energy can be saved without violating comfort conditions. These time periods are defined by an occupancy parameter which is a binary-valued Schedule. The value of one (1) is used to indicate that, the respective space, is occupied and the value of zero (0), otherwise. Generally, occupancy is a parameter which is



difficult to estimate. In order to predict future occupancy values statistical data obtained by the data warehouse can be used referring to past occupancy schedules. Consequently, either past occupancy data or forecasts based on these data could be used.

### **A.2 Internal gains schedules**

Operating building equipment (computers and electrical equipment) as well as the presence of people acts as internal thermal sources (air and surrounding internal surfaces). Since this total amount of thermal energy, is a general nonnegative number varying with time, it can be represented by a schedule (a time varying nonnegative continuous function). This schedule is called internal gains schedule and is assigned to every building space. Usually the internal gains are estimated based on the number of people being inside a building space and the operational schedules for equipment in the same space.

### **A.3 Uncontrollable device schedules**

During the building operation the schedules of certain devices are determined only by the user of these spaces. Furthermore some devices cannot be controlled by the Baas system as there is no appropriate interface installed. The above devices can be grouped together as uncontrollable devices. Their operations are determined by respective schedules (SCH). The schedules of these devices are vector functions containing:

- A binary (1=on/ 0=off) state time function.
- Operation mode time functions.

These include time function describing how the operation mode characteristics of these devices change with time. For example if an HVAC belongs to the group of uncontrollable devices, the operation mode characteristics may refer to heating and cooling set-points, fan speed and other parameters defining its operation.

### **A.4 Controllable device schedules**

Devices controlled by Baas system belong to the class of controllable devices. As in the case of uncontrollable devices the operation schedule of controllable devices play important role in the calculations performed during simulation APO calls. The schedules of controllable devices are determined by either model-based or rule-based control decisions mentioned in the previous sections. These schedules refer to the same (binary state and operation mode) time functions as the schedules of uncontrollable devices.

### **A.5 Effects of User Behaviour**

The second significant uncertainty with respect to forecasting and simulation is the modelling of the actions of users in the building. There are two important aspects to be explicated within this context: the prediction of occupancy patterns, so that the system can have an estimate on when occupants will be present and when there will be cooling and/or heating needs that have to be satisfied; and the effect of user actions with respect to uncontrollable devices in the building (e.g. opening and closing windows). Creating forecasting models for user-behaviour requires models for estimation of the effects of these parameters. Occupancy patterns (schedules) can be estimated utilizing a number of information sources:

- Reference Schedules as defined for particular building typologies in standards, or as user-input for the model (e.g. in office buildings, occupancy schedules are typically the working-hour schedules)
- Measured or inferred occupancy patterns: utilizing, if available, occupancy sensors installed in the building (if this information is available), and occupancy inference

methods (occupancy-existence reasoning modules) that use other measurements (e.g. energy meters, HVAC operation in a room) to infer occupancy in building spaces.

- Data available from scheduling systems, indicating use of certain rooms (e.g. booking system for a conference room).

Fusion of past, reference and forecast data can help provide forecasting of occupancy patterns. These patterns can then become available (“consumed”) through the integration layer to other service components requesting them (e.g. simulation models). For the needs of BaaS the detailed prediction of occupancy patterns is not desired or possible (e.g. it is not possible to predict when a user will move out of his room and come back again). What is the (partial) outcome of this effort is a model that is capable of predicting the statistically-average occupancy patterns to be used, e.g. for simulation purposes.

The second aspect to be considered in this context is the prediction of energy-influencing user actions (i.e. behaviour). As before, it is not possible or desired to have accurate predictions of the user actions. Rather it is useful to have a statistically correct model for their behaviour. In that sense, the second part in this task is the modelling of the user-behaviour. We intend to approach this aspect as follows: we separate the energy-influencing actuating resources in two categories: controllable (by the Building Energy Management System) and uncontrollable (but influenced by user actions) – by way of an example, the opening and closing of the windows is an uncontrollable resource. In the first category the control logic creates the actuation commands, whereas in the second category user actions are the actuating components. In this view user behaviour can be represented as control components that implement the actions for the uncontrollable resources. The models for user behaviour are then control actions triggered by the user. This is especially important as it allows a uniform treatment of control actions. Users can act rationally or (more often) irrationally, rules (or generally using inference methods) can be used to represent user actions. Factual data (measurements) available from the monitoring system can be used to dynamically improve the models and represent (individualized) rational or irrational behaviour patterns. User effects can be incorporated to the uncontrollable schedules defined above.

## **B. Other simulation parameters (OSP)**

Every building simulation process requires timing information in order to be performed as well as knowledge of the required output parameters. This type of information does not belong to any previous categories and can be grouped to a new class called other simulation parameters (OSP). More precisely OSP parameters include:

- Simulation start time (defined by: month, day, hour and minute).
- Simulation end time (defined by: month, day, hour and minute).
- Simulation inter-sample time interval (in minutes).
- Preferred outputs (for example: temperature, humidity, energy demands).

The values of these parameters are either defined by the user of the APO services or the control and fault detection processes. In addition, to the items above, we include in the OSP category, the desirable outputs that successful invocation of the calculation methodology should return.

## **C. Weather data**

As the calculation performed using physical models require the knowledge of boundary conditions, in buildings the simulation calculations require weather data values in order to be executed. Most of these data are provided by weather files which contain measurements obtained by weather stations. These are called weather file data. However there are weather

parameter values which are not provided by a weather file and have to be estimated [12]. These belong to the estimated weather data category. As the formats of weather files may differ some weather files may not contain the values of certain weather parameters. Similarly to the values of the material some of the weather data values can be classified as fundamental if their values cannot be obtained using other weather parameter values and derived otherwise. In conclusion, weather data values are essentially sampled values according to the simulation inter-sample time intervals and are obtained either by using past values from the data warehouse, or by a weather forecast. The creation of weather files will be further elaborated in WP4.

### 3.2.3 Access to dynamic data: co-simulation

Dynamic Data Schedules can also be classified into three broad categories:

1. **Past data (P).** (Source: Data Warehouse)

Includes factual data obtained from in building sensors, or other historical data obtained during actual building operation, gathered from the present time instant and backwards.

2. **Forecast data (F).** (Source: External Services, Forecasting Modules)

Forecast data refer to predicted data obtained from external services or computed from forecasting modules. These data include weather predictions (obtained from a weather prediction service), occupancy forecasts (obtained from a room scheduling system), etc.

3. **Reference data (R).** (Source: Data Warehouse)

Reference data refer to synthetic data that is used in the absence of real measured data or forecasts. These can be reasonable default (reference) values to be used in the simulation when no other information is available. These might be obtained using statistical aggregations of past data (e.g. Meteorological weather data) or can be inferred based on the building typology or other reasonable assumptions (e.g. occupancy schedules follow the working schedules). Obviously this synthetic data represents “average” schedules to be used in place of real data when these are not available. An example could be when occupancy sensors are not installed in the building (or occupancy information cannot be indirectly inferred from other sources), reference occupancy data can then form reasonable substitutions. In the design-phase utilization of calculation methodologies all dynamic data used are reference data.

Establishing a link between the calculation methodology and the external data sources, so that these dynamic schedules can be incorporated in the simulation, is of paramount importance for achieving all the tasks mentioned above (irrespective of the data source). This functionality is especially desirable also for the testing and design of control strategies. Creating therefore an interface to enable two-way communication, at each time step of the simulation, is essential, and this functionality we refer to as co-simulation. By way of example, in EnergyPlus the BCVTB (Building Control Virtual Test Bed) [31] component can be used to implement co-simulation capabilities. In TRNSYS historically it was Type155, MATLAB connection was in charged to create the connection with optimizers, but nowadays, BCVTB is also able to work with TRNSYS allowing using the same methodology as the one used with Energy+. In all cases, the communication interface, describing the objects to be exchanged during co-simulation, can be defined using a properly formatted XML file. Within BaaS, it is the intention to support within the data structures this co-simulation linking mechanism and to provide ways to generate these interfaces.

Access to static and dynamic data is made available using the signal abstraction at the APO Kernel Layer, establishing connection to the communication layer. The details on how this will be achieved within BaaS are presented in BaaS D5.1, where the Signal Handler and the connections to the data layer are explicated.

In view of this dynamic data categorization, Table 5 summarizes the data access requirements for the APO Tasks. As can be seen in the Table, depending on the Task requirements can differ so this establishes the need for a generic link mechanism between data and simulation.

**Table 5: APO tasks and their simulation functional requirements**

APO Simulation Task	Dynamic Data			Static Data
	Past	Forecast	Reference	
Model calibration	X			X
Components validation	X			X
Energy performance estimation	X		X	X
Energy performance forecasting	X	X		X
Virtual sensors	X	X	X	X
Control design	X	X	X	X
Control design optimization	X	X	X	X

### 3.2.4 Conclusions

Apart from static BIM information regarding the building geometry, the construction material properties and the characteristics of the building systems as well as other simulation parameters (OSP), the above simulation types require dynamic data of the previous three types belonging to the following **categories** (analysed Dynamic Data):

1. Occupancy data (OCC)
2. Internal gains data (ING)
3. Weather data (WTH)
4. Uncontrollable device schedule data (UDS)
5. Controllable device schedule data (CDS)

Table 6 classifies the data requirements of the simulation types called by the APO services with respect to the previous categories and types classification. Simulation tools (or engines) which require data may reside to a remote simulation server and called when needed.

**Table 6: Simulation Data Requirements and their data sources**

Static Data Requirements	Data Sources		
<b>BMD: Geometrical</b>	BIM Server		
<b>BMD: Material</b>	BIM Server		
<b>BMD: Systems</b>	BIM Server		
<b>OSP</b>	APO kernel		
Dynamic Data Requirements	Data Sources		
	Past	Forecast	Reference
<b>WTH</b>	DW	External Service	DW
<b>SCH: OCC</b>	DW	External Service	DW
<b>SCH: ING</b>	BIM Server	DW	DW
<b>SCH: UDS</b>	BIM Server	DW	DW
<b>SCH: CDS</b>	BIM Server	DW	DW

### 3.3 APO layer integration requirements

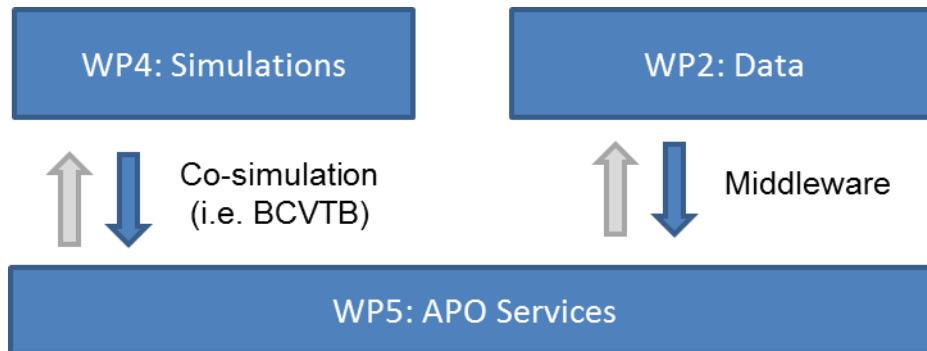
In order to illustrate the interaction between the prediction tools developed in WP4 (simulation models) and the APO services, a simplified model-based control design optimization task is presented, addressing the optimization of night setback of the heating system for the Sierra Elvira School.

Here, the daily heating schedule (start/stop time) is governed by a fixed schedule, thus can be further optimized utilizing in-building sensor measurements and weather forecasts, in order to avoid overheating problems during noon. The optimization task attempts to minimize the energy consumption while maintaining indoor thermal comfort, by regulating the amount of water sent back to the boiler for heating.

In the remaining sections, the properties of the simulation model required for the task are described. The contents of this section, along with the definition of the optimization setup establish the interplay and interdependencies between WP4 and WP5.

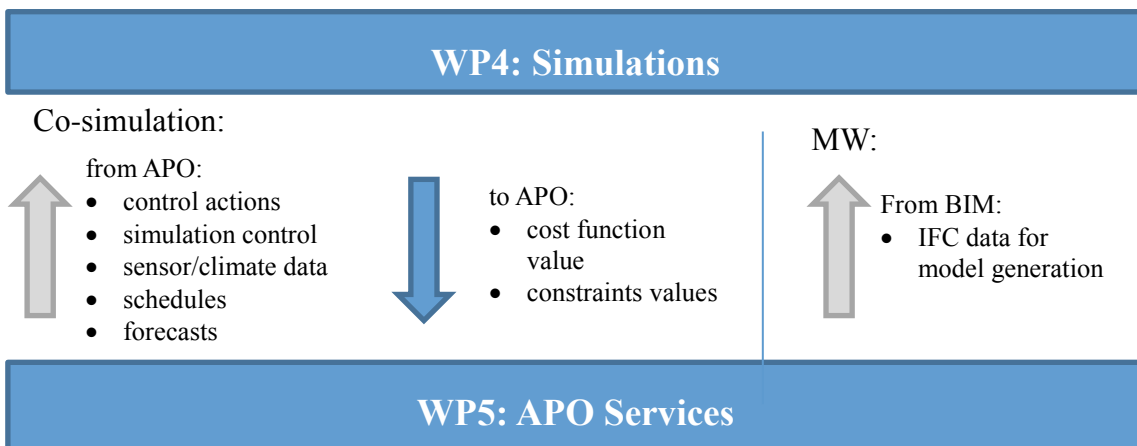
#### 3.3.1 Data flow

The main interaction/data exchange scheme between WP4 and WP5 is shown below. Here, the dynamic data schedules required by the simulation model are retrieved by the repositories of WP2 (BIM, DWH and External Services) and are provided to the simulation through the APO kernel, using proper interface. Subsequently, the retrieved data are injected to the simulation model through the co-simulation interface.



**Figure 15: Interconnection principle, i.e. WP4, WP5 and WP6**

It is evident from the figure that no direct access between the components of WP4 and WP2 is established, but the APO kernel is responsible for providing WP4 with the necessary inputs. A closer look in this data exchange scheme is shown in the following figure. Here, in each timestep of the simulation, the co-simulation module injects the control actions, historical data, reference schedules and forecasts to the model, while forwards cost function and constraints values to the APO service that requested the prediction task. Independently, a separate request to the BIM (performed only once), carries the IFC data for the model generation.



**Figure 16: Data exchange via co-simulation**

### 3.3.2 Integrating simulation capabilities as services to the APO layer

Although WP4 focuses on generating and deploying detailed thermal simulation models as surrogates of the process (building), other modelling approaches, corresponding to various APO services implementations have to be able to be incorporated to BaaS system (e.g. state-space models, black-box data-driven models, etc.).

To that direction, to allow smooth expandability of the system, the APO layer should be able to host and invoke various types of process models, without providing support on the implementation of these models. In other words, the *simulation models of the building components* (like PVs, HVAC systems, etc.) should not be “hardcoded” to the kernel, but the kernel should be able to invoke process models from the available models library (Figure 17).

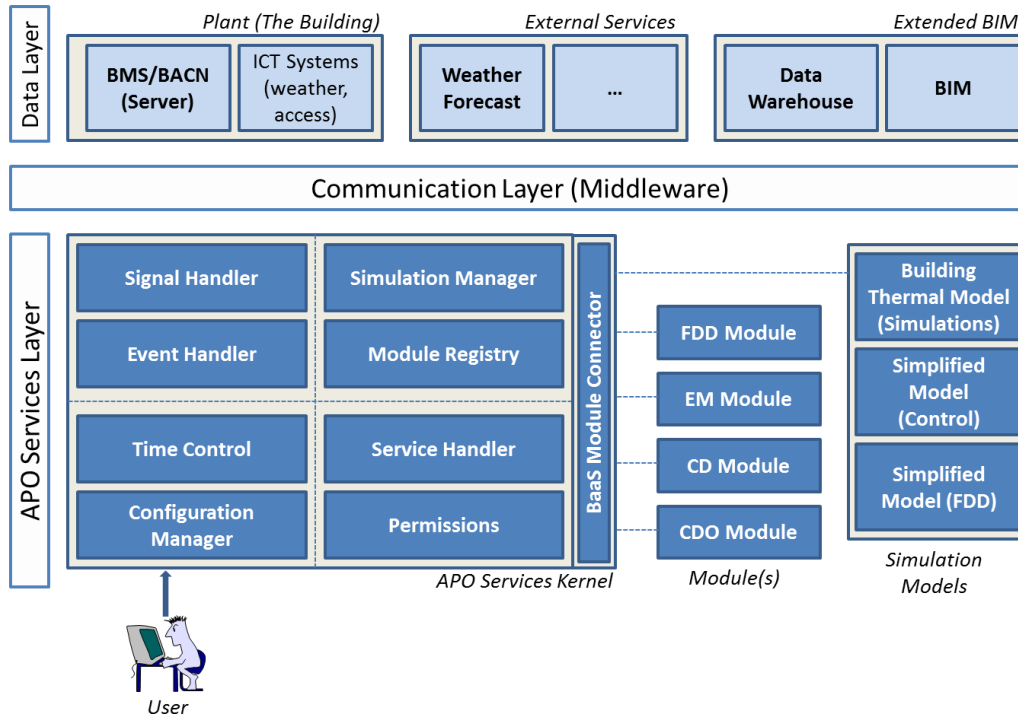


Figure 17: Core components of the APO Layer

### 3.3.3 Simulation Manager

In view of these design requirements for the integration of models within the APO layer, the simulation manager is developed. The simulation manager is responsible for providing a fully functional *simulation model* (whole building or component) to any APO service that requests it. In other words, any APO service can request a simulation run on a specific model for a specific time-period and requesting as output any KPI, and the simulation manager is responsible to fulfil the task, using the simulation and co-simulation abstract components. In such case, the simulation manager creates a new simulation object to perform the assigned simulation task, inside which a co-simulation object is instantiated to manage the input-output schedules.

#### 3.3.3.1 Simulation abstract component

Here, the static part of the model (e.g. information on geometry and materials) has been generated and the different models are available in the models library. For a simulation scenario to be formulated, the following (high-level) components are required:

- **Building model:** the simulation object has to be correlated to a specific building model, according to the specific task and the corresponding simulation scenario.
- **Simulation Outputs:** the specific KPIs requested by the parent APO service as outputs are defined and selected among the output options of the model.
- **Co-simulation binding:** a mapping between the co-simulation object ports and the inputs-outputs of the model is constructed here
- **Other simulation parameters:** a multitude of simulation parameters adjusting the simulation setup are selected, including the starting and ending time of the simulation, the simulation time-step, etc.

### 3.3.3.2 *Co-simulation abstract component*

As thoroughly discussed in the previous section, several input schedules have to be defined and injected to the model to perform a simulation, while a number of KPIs requested by the parent APO service as outputs (e.g. energy consumption and thermal comfort) have to be selected among the possible outputs offered by the model. This is achieved with the use of the co-simulation setup. Here, the simulation manager identifies the specific schedules that will be used by the simulation along with the various functions required and a co-simulation object is instantiated inside the simulation object, including the following methods:

- **Input schedules:** the schedules that will be forwarded by the co-simulation object to the simulation object. Here, apart from uncontrollable device schedules, weather data and occupancy predictions, the input schedules may come from the output of other modules. A typical example of the latter case is the interconnection between the co-simulation and the controller modules. Here, the co-simulation object outputs the current states of the building and the controller generates control inputs for specific actuators, passed as input schedules to the co-simulation;
- **Output schedules:** the final output schedules of the simulation. For example, for an energy auditing APO service, the simulation output would be the total energy consumption, while for a control design optimization task, would be the performance – in terms of energy consumption and user comfort of a candidate controller.

### 3.3.4 *Configuration*

Even though the APO kernel manages to regulate the interactions between the different components required for a simulation model call, a configuration process precedes to define the details of the components involved, i.e. the specific model used, the simulation outputs, the input schedules and the specific functions (controllable device schedules – controllers) to be utilized. Once all components are configured, the model is available for invocation from the respective APO modules.

Once the overall process is configured, the APO kernel acts as an interoperability core component to facilitate communication and support the business processes and data access requirements.



---

## 4 Model View Definitions (MVDs) for thermal simulation

### 4.1 Method for defining and developing interfaces - introduction

The present document focuses on defining the functional requirements for the simulation models, as well as providing an overview of the modelling approach within BaaS project, i.e. using thermal simulation tools. As the BaaS platform aims for elevated expandability properties, a clear interface has to be defined, in order to provide stakeholders the option to develop and deploy to the system new models. Within BaaS IFC data available from the BIM repository are utilized for the interface definition.

To that direction, illustrative examples using TRNSYS simulation software are included in the present document defining the core functional requirements, which also hold for other simulation software, like EnergyPlus or Modelica for example. Note here that the final adaptation of the information required for defining the simulation models will require an adaptation/processing task for all simulation software, but the functional requirements will be the same for all target software.

This raises the need for defining a subset of the IFC, containing only the information required for defining this interface (work done by the MVDs). In addition to this, a set of default properties for missing information can be determined, as well as additional relationships between IFC entities that simplify the specific task. Such an interface, apart from providing the requirements for defining simulation models, can be utilized for consistency checking and fault detection on acquired data, as well as for definition of software tools developed for the task, required to support only the interface definitions and not the whole IFC.

Towards implementation and testing, at the end of the project the interface requirements will be automatically included in the whole BaaS system. Within the definition and development processes several helper applications will allow the testing of the functionality of the developed code, i.e. standalone BIMServer, software applications run from IDE (integrated development environments) like Eclipse, MVD tools, IFC exporters and viewers as well as text editors.

As an example, the Annex A of this D4.1 document presents the result of a filtered query to an IFC file that permits the creation of a simulation model without reading the complete IFC design.

### 4.2 Model View Definitions

Since the IFC model contains a significant amount of information covering most functional requirements during the lifetime of a building, a “sorting” process has to proceed, where all the information not required by the simulation model definition task will be excluded from the interface definition. For the remaining information, two additional tasks are required:

- define the rules for exception handling of any kind, e.g. define default values for missing data or inconsistencies;
- define obligatory requirements that have to be fulfilled.

The final set of “rules” are called “exchange requirements”, defining the combination of what will be exchanged in which way and communicate these requirements to everyone. Thus, except from properly defining these requirements it is essential to communicate these requirements in a structured and standardized way; within BaaS this “guidebook” is defined using Model View Definitions (MVD).

According to buildingSMART website<sup>11</sup>, "An IFC Model View Definition defines a legal subset of the IFC Schema (being complete) and provides implementation guidance (or implementation agreements) for all IFC concepts (classes, attributes, relationships, property sets, quantity definitions, etc.) used within this subset. It thereby represents the software requirement specification for the implementation of an IFC interface to satisfy the Exchange Requirements.

The Model View Definition XML format (mvdXML) captures the following:

- Exchange Definitions indicating import and export scenarios for each model view
- Concept Roots indicating entities of the IFC schema included within the model view
- Concept Nodes indicating rules for valid attribute values and referenced types
- Concept Templates which may be re-used across related entities
- Exchange Requirements indicating mandatory, optional, or excluded concepts for each exchange."

An MVD can be described as an intermediate state of data structure in-between the original IFC file and the result of a certain query. The physical format of the MVD could still be an IFC file or an mvdXML<sup>12</sup> file, which is the method to publish the concepts and associated. The main task of an MVD is to filter the IFC data in order to reduce the amount of information and focus on data related to a defined subject, i.e. structural data, HVAC data, etc. The additional important role of MVD is allowing consistency checks of the data, exception handling and filling of "gaps" in data (as far as possible). MVDs are defined and published using the "ifcDoc"<sup>13</sup> tool. For more technical details on the selected standards, please refer to <http://www.buildingsmart-tech.org/specifications/ifc-view-definition/fm-handover-aquarium/fm-basic-handover>

### 4.3 Definition of "exchange requirements"

In view of the discussion above, it can be expected that several exchange requirements, corresponding to different functional tasks, will be defined within BaaS project and eventually combined to the overall BaaS MVD. Since this is an on-going process, it is not possible to define all exchange requirements beforehand, but the BaaS MVD will be constantly updated, reflecting new insights coming from practical implementation. Nevertheless, it is clear for the simulation models definitions, that the defined exchange requirements will be "consumed" in various points of the model development:

- **Building design phase:** the building architect should take into account the exchange requirements when designing the building in a CAD tool (e.g. simulation software requires space definitions). In addition to this, IFC export tools (e.g. Revit or Constructivity) should also have the same information, but defining and requesting support for custom MVDs from commercial tools can be a time-consuming task.
- **Operational phase:** the exchange requirements will provide the guidelines in which queries to the BIM will return "meaningful" information for the task, but also will assist towards defining the interface for defining simulation models.

<sup>11</sup> <http://www.buildingsmart.org/>

<sup>12</sup> <http://www.buildingsmart-tech.org/blogs/msg-blog/mvdxml-1.0-released>

<sup>13</sup> <http://www.buildingsmart-tech.org/blogs/msg-blog/ifcdoc-the-new-mvd-development-tool>

---

For our BaaS cases of study, the MVD's are required to create accurate simulation building and component models. Those must contain every parameter needed by the selected simulation software as well as the relations among components:

- Physical connections to the previous and following component. In this category, simulation software doesn't need the physical placement of the components unless they are affected by the solar radiation or wind effects as can be in wall cases. Most of the components only need the definition of which is the component that delivers them a thermal fluid, in what conditions is the fluid delivered and to which component will be sent after a determinate change of energetic state caused by the idiosyncrasy of the studied element.
- Parameter determination of the studied element. Once that the element is connected to the surrounding ones, the processes occurring in it should be defined by a set of efficiencies, and parameters that describe them. These parameters should be directly taken from the IFC file or easily calculated from data found in it.
- Relationship or belonging of each one of the components to a simulated zone. The occupied zones, defined as a single unit or a group of them, are in the end the link between the thermal systems installed in a building and the thermal characteristics of a determinate zone inside a complete studied building.

The following subchapter 5.10 represents the connection between components that integrate the IFC HVAC domain and the relations with the TRNSYS types that simulate them. As it is explained there, most of the parameters needed for the simulation are included in the IFC file while some of them are written as a default value depending on the representing component.

## 5 Data acquisition

For detailed information about BIM and IFC refer to the deliverables D2.1 "Data Warehouse Requirements and extended BIM specification", for description of data handling and exchange see deliverable D2.3 "BIM repository and associated methods and tools".

Important notice: The following sections contain the description of simulation model development mainly using the tool TRNSYS, where this tool just serves as an example (and the author is more familiar with the handling of this tool). As explained in section 2.3 the simulation model development can principally be transferred to other simulation tools like Energy Plus or combinations of different tools within a co-simulation framework, due to the similar underlying model structure and due to the modularity of the developed simulation models.

### 5.1 Actual state of versions and tools (summary)

The underlying main structure is defined by the building data model according to the definition of IFC schema. The current version (IFC4) is an ISO standard ISO 16739 and is the evolution of the previous IFC2X3 version. Here, the data is physically stored in a STEP file (readable with a text editor) using EXPRESS data definition language (defined as ISO 10303-11). In addition to the IFC-EXPRESS specification an ifcXML specification is published as well, following the XML document structure.

The IFC-file usually is generated by the export of a software tool, mostly CAD systems with implemented interface (currently IFC2x3), such as 'Revit'. Actually there is no software available implementing an IFC4 export function besides a new tool called 'Constructivity', but still with limited IFC4 support (in fact, IFC2x4 Release Candidate 4). This is the main reason that the example queries in the present document are in IFC2X3; once stable versions of the tools are provided for IFC4 the queries will be adapted if necessary

The text file can also be changed "by hand" using an editor and following strictly the STEP language requirements. Several tools offer the possibility to generate a 3-dimensional view of the IFC file content, such as 'Solibri viewer'. Additionally there are tools to edit or check IFC files without using a CAD software.

The BIM server facilitates the storage of and access to IFC files in a data base and furthermore the data management including version control and multi-user support. The 'BiMserver project' around TNO and Eindhoven University provides an online access to their demosever as well as the download of local servers. The server is based on Java-code and open source. Current stable version (1.1) implements an IFC2x3 core, but the server has been adapted to IFC4 within BaaS project.

The BIM server also provides simple queries of the building model, i.e. target selected information out of the model content. The query language is Java. It is also possible to query the model with a standalone tool bypassing the user interface of the BIM server. This can be realised by Java code (environment e.g. Eclipse), using the libraries of the BiMserver. The code than gets access to the BiMserver, loads (deserialises) an IFC file into an object and can be used to define different logic for certain data request. The use of the BIM libraries and the data model enables a comfortable handling of the data but is limited to the state of completeness of the implantation regarding server and the IFC version.

In order to do one step back, the queries can also be realised using direct access to the IFC file without having a BIM server (or libraries) involved. Since IFC is using the STEP schema, there are several tools available for parsing STEP files (direct access to text/STEP file).

Within the 'buildingSMART' project, a new specification of scheme was introduced called 'Model View Definition MVD'. A MVD can be described as an intermediate state of data structure in-between the original IFC file and the result of a certain query. The physical format of the MVD is still an IFC file or alternatively an xml file. The tool developed to define and support MVD definitions is called 'ifcDoc', is valid from IFC4 onwards.

## 5.2 Sources of information, references

The most recommended source of information related to IFC, BIM, etc. is represented by 'buildingSMART International', describing itself as "*A neutral, international and unique non for profit organisation supporting open BIM through the life cycle.*"

**Table 7: Sources of information, references**

Subject	what	where
All	BIM, open BIM, certification, tools, references, MVD	<a href="http://www.buildingsmart.org/">http://www.buildingsmart.org/</a>
EXPRESS	data modeling language	<a href="http://en.wikipedia.org/wiki/ISO_10303-11">http://en.wikipedia.org/wiki/ISO_10303-11</a>
STEP	STEP file, data exchange form	<a href="http://en.wikipedia.org/wiki/ISO_10303-21">http://en.wikipedia.org/wiki/ISO_10303-21</a>
STEP+IFC4	tools, examples	<a href="http://www.steptools.com/support/stdev_docs/ifcbim/index.html">http://www.steptools.com/support/stdev_docs/ifcbim/index.html</a>
IFC	releases	<a href="http://www.buildingsmart-tech.org/specifications/ifc-releases/summary">http://www.buildingsmart-tech.org/specifications/ifc-releases/summary</a>
IFC2x3	implementation guide	<a href="http://www.buildingsmart-tech.org/downloads/accompanying-documents/guidelines/IFC2x%20Model%20Implementation%20Guide%20V2-0b.pdf">http://www.buildingsmart-tech.org/downloads/accompanying-documents/guidelines/IFC2x%20Model%20Implementation%20Guide%20V2-0b.pdf</a>
IFC4	reference	<a href="http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm">http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm</a>
BiMserver	downloads, wiki, sources	<a href="http://code.google.com/p/bimserver/">http://code.google.com/p/bimserver/</a>
BiMserver	support, community	<a href="http://support.bimserver.org/bimserver">http://support.bimserver.org/bimserver</a>
MVD	ifcDoc Tool	<a href="http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool">http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool</a>
MVD	IFC Implementation Guide: Space boundaries for thermal analysis	<a href="http://www.buildingsmart-tech.org/downloads/accompanying-documents/agreements/IFC2x3-space-boundary-implAgreement-2009-09-17.pdf/view">http://www.buildingsmart-tech.org/downloads/accompanying-documents/agreements/IFC2x3-space-boundary-implAgreement-2009-09-17.pdf/view</a>
Tool	query language: BIMql	<a href="http://bimql.org/">http://bimql.org/</a>
Tool	export of IFC4-files	<a href="http://www.constructivity.com/">http://www.constructivity.com/</a>
Tool	IFC viewer	<a href="http://www.solibri.com/">http://www.solibri.com/</a>

Tool	GUI for EnergyPlus: Simergy (strength in HVAC)	<a href="http://simulationresearch.lbl.gov/projects/gui">http://simulationresearch.lbl.gov/projects/gui</a>
TRNSYS	simulation tool	<a href="http://www.transsolar.com/_software/docs/trnsys/trnsys_uebersicht_en.htm">http://www.transsolar.com/_software/docs/trnsys/trnsys_uebersicht_en.htm</a>

### 5.3 IFC-structure regarding thermal simulation

For further information about IFC structure and data handling see deliverables D2.1: Data Warehouse Requirements and extended BIM specification and D2.3: BIM repository and associated methods and tools.

#### 5.3.1 IFC schemas and domains

The specification consists of a schema defining data types, along with common concepts indicating use of data types for particular scenarios. These common concepts are applied at entities having specific use. Such concepts also form the basis of Model View Definitions, which are supplementary specifications that adapt the scope and rules of this schema for targeted domains within the building industry.

Each concept template defines a graph of entities and attributes, with constraints and parameters set for particular attributes and instance types. Various entities within this schema reference such concept templates and adapt them for particular use according to parameters.

The schemas to be studied surround mainly core schemas, shared schemas, domain schemas and resource schemas. Additional schemas used for auxiliary data (e.g. relations, units etc.) are contained in shared and resource schemas. The various entities are assigned to domains as can be seen in the next figure. The domain specific data schemas contain final specializations of entities as shown highlighted in blue. Entities defined in this layer are self-contained and cannot be referenced by any other layer. The domain specific layer organizes definitions according to industry discipline. The figure also contains the contribution of consortium partners using the logo and the limitation of IFC domains to be studied by using a strikeout sign, i.e. focussing on

- Electricity domain (exploded by CARTIF) is formed by:
  - 11 components.
  - 9 were previously discarded
- Control domain (exploded by TUC):
  - 9 components (no one previously discarded)
- HVAC domain (exploded by HON & Fraunhofer):
  - 29 components
  - 5 previously discarded

The chosen components must be nowadays re-filtered in order to eliminate information not needed for simulations.

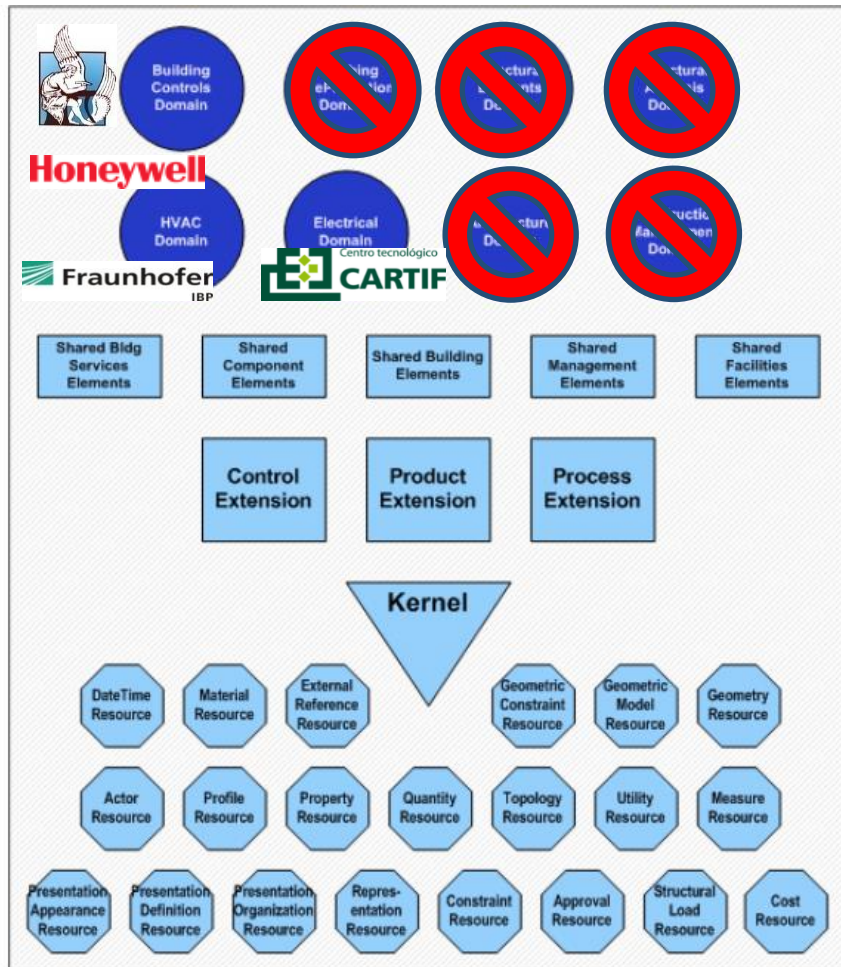


Figure 18: IFC domains and contribution of partners

### 5.3.2 IFC data types

Mainly we can distinct five different groups of data types which represent different data structure and treatment, i.e.

- Enumerations (suffix "TypeEnum"): construct that allows an attribute value to be one of multiple predefined values identified by name
- Types (suffix "Type"): basic information construct derived from a primitive, an enumeration, or a select of entities
- Entities (occurrences of types): class of information defined by common attributes and constraints as defined in ISO 10303-11
- Property sets: collection of information that is dynamically defined as a particular entity instance
- Quantity sets: collection of measurements of a scope-based metric, specifically length, area, volume, weight, count, or time

The following tables contain the result of the study of IFC domains described in section 5.3. The tables are divided in terms of relevance for thermal simulation. The optional entities have to be checked in case of occurrence. Each listed entity represents all types of available data, i.e.

Enumerations, Types, Entities, Property Sets and Quantity Sets. Example: the table contains the entity "IfcBoiler" which stands for the relevance of the following IFC-data:

- IfcBoiler: the occurrence of the boiler itself (Entity)
- IfcBoilerType: basic information construct
- IfcBoilerTypeEnum: Enumeration defining the typical types of boilers
- Pset\_BoilerX: Property Sets related to IfcBoiler and IfcBoilerType (X as wildcard for certain names defined for Boiler)
- Qto\_BoilerX: Quantity Sets related to IfcBoiler and IfcBoilerType.

Where the IfcBoilerType is used to define a boiler type specification indicating the specific product information that is common to all occurrences of that product type.

**Table 8: List of relevant Entities - HVAC**

<b>relevant</b>		
IfcAirToAirHeatRecovery	IfcEvaporativeCooler	IfcPump
IfcBoiler	IfcFlowMeter	IfcSpaceHeater
IfcChiller	IfcFan	IfcTank
IfcCoil	IfcHeatExchanger	IfcUnitaryEquipment
IfcCoolingTower	IfcHumidifier	IfcValve
<b>optional</b>		
IfcAirTerminal Box	IfcCompressor	IfcDuctFitting
IfcAirTerminal	IfcEvaporator	IfcDuctSegment
IfcBurner	IfcCondenser	IfcDuctSegment
IfcCooledBeam	IfcPipeSegment	IfcTubeBundle
IfcDamper	IfcPipeFitting	IfcEngine
<b>not relevant</b>		
IfcVibrationIsolator	IfcFilter	IfcMedicalDevice

**Table 9: List of relevant Entities - Control**

<b>relevant</b>		
IfcActuator	IfcAlarm	IfcController
IfcFlowInstrument	IfcSensor	IfcUnitaryControlElement
<b>optional</b>		
IfcSwitchingDevice		
<b>not relevant</b>		
IfcProtectiveDevice	IfcProtectingDeviceTripping	

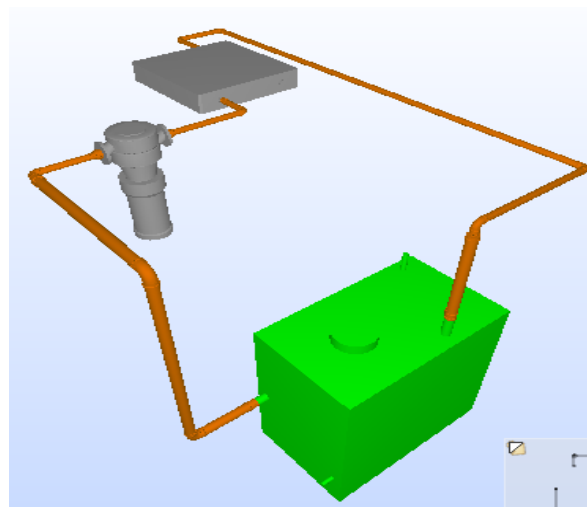


**Table 10: List of relevant Entities - Electricity**

<b>relevant</b>		
IfcAudioVisualAppliance	IfcElectricGenerator	IfcElectricTimeControl
IfcElectricAppliance	IfcElectricMotor	IfcSolarDevice
IfcElectricFlowStorageDevice		
<b>optional</b>		
IfcLamp	IfcTransformer	IfcCommunicationAppliance
IfcLightFixture		
<b>not relevant</b>		
IfcCableCarrierFitting	IfcCableSegment	IfcMotorConnection
IfcCableCarrierSegment	IfcElectricDistributionBoard	IfcOutlet
IfcCableFitting	IfcJunctionBox	

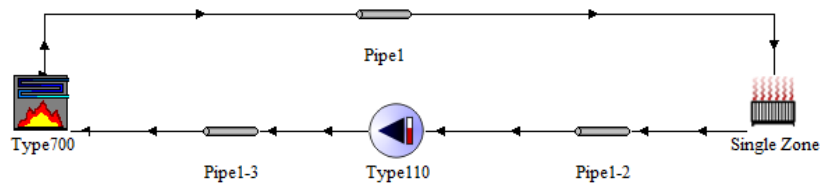
#### 5.4 Simulation model development

The main goal within this project is the assessment of energy demand/consumption of buildings. Therefore the knowledge of the energy demand of building is required. The building is represented by zones which are assigned by terminal elements of the energy supply system (e.g. radiator) where at the end the evaluation of energy generation (e.g. boiler) results the final or primary energy demand. The following figure contains a simple model of the energy system, i.e. a heating circuit containing radiator, pump and boiler. The model is displayed in a IFC viewer (Solibri) and generated by a CAD tool (Revit).



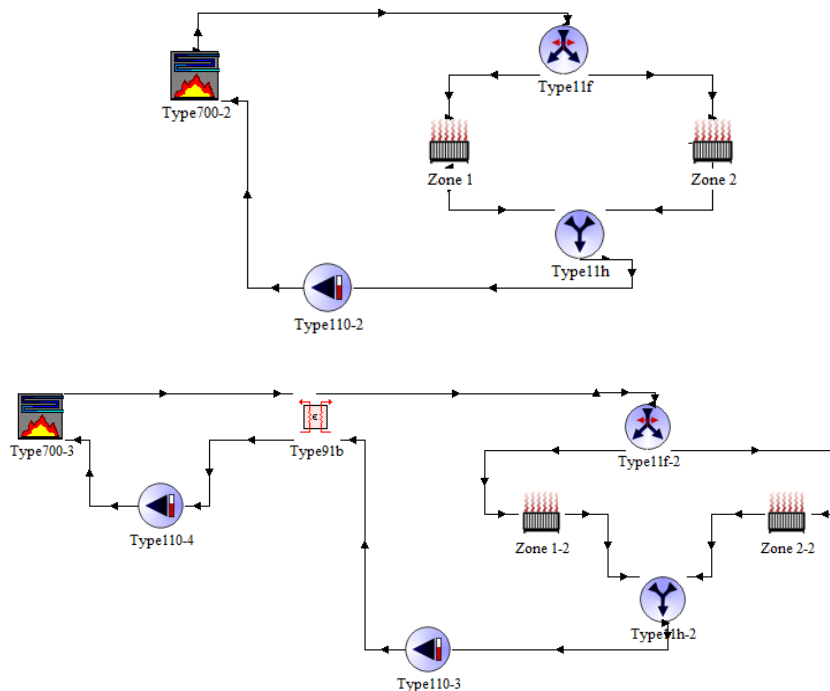
**Figure 19: Simple model of boiler/pump/radiator circuit (IFC viewer)**

The corresponding model for thermal simulation is displayed in the next figure as simulation model. TRNSYS offers several components (called "Type") for modelling pump and boiler as well as the pipes in-between and the "Single Zone" for demand side and radiator. The model contains sample Types, within the project alternative Types will be studied.



**Figure 20: Same circuit as model in TRNSYS**

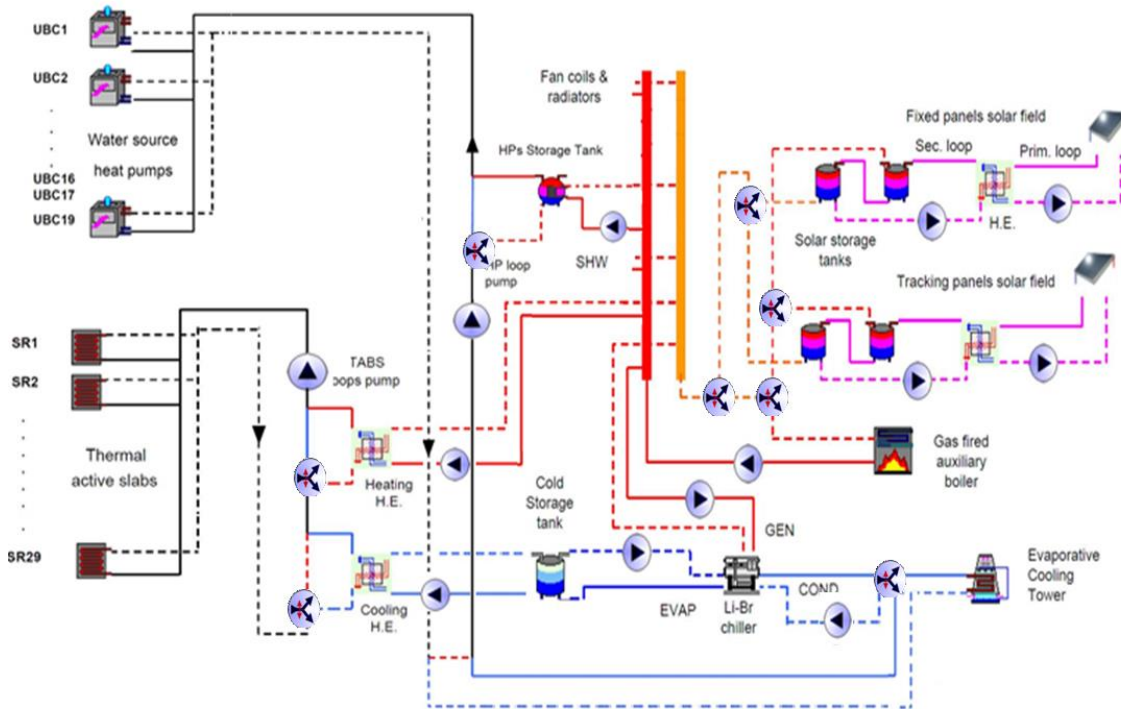
The next figures show system models of advanced systems, i.e. the extension of the previous model by two radiators installed in parallel, hydraulic separation and an additional heat exchanger.



**Figure 21: Advanced systems (TRNSYS) - parallel demand**

The most advanced model contains parallel demand, hydraulic separation, intermediate storage and multiple generation. The model is developed for one of the test-beds of the BaaS project: the CARTIF-Building in Valladolid, Spain. This building is equipped with the most complex energy systems compared to the other test-beds and pilot buildings. In order to give a brief outline of the equipment the model contains

- two different heat emission systems to the zones, i.e. water source heat pumps and thermal active slabs (left in the figure)
- generation: gas boiler, absorption chiller, cooling tower, solar collectors (right)
- heat and cold storage tanks (centre), solar storage tanks
- heat exchanger and distribution elements, i.e. pumps, pipes, flow diverter, tee-pieces



**Figure 22: Advanced system - multiple generators**






As can be seen in the presented models, the system generally consists of one or several terminating emission components (mostly placed in a building zone), one or multiple generation systems and a circuit, represented by distribution elements. Additional components like storage tanks or heat exchanger can also be part of the circuit. The linking of the components by the distribution system define uniquely the relations between the subsystems, e.g. which generator serves specific zones, with special regard to distributors (flow diverter) and collectors (tee-piece).


Thus, a circuit model that provides parallel demand and multiple generation is able to cover all requirements of the building heating systems for the test-beds and pilot buildings within the BaaS project.

The development of the presented simulation models is mainly based on the heating system. Other systems are cooling systems and air handling units. In case of water based cooling (e.g. thermal active slabs) the general principle of a heating circuit can be directly transferred since emission/distribution is equivalent and storage/generation similar.

The following table gives an overview of the TRNSYS types used for the model of the generic heating and cooling circuits, i.e. generation components for heat and cold, distribution elements and an emission component as link to the zone demand.

**Table 11: TRNSYS types for components in heating and cooling circuits**

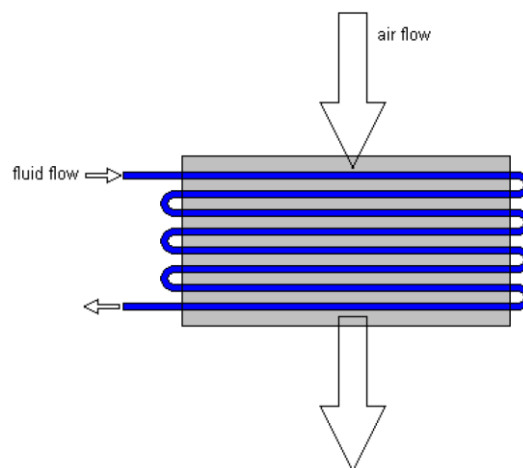
Component	Icon	Description
Auxiliary Heater w/Proportional Control	 Type659	Type659 models an external, proportionally controlled fluid heater. External proportional control (an input signal between 0 and 1) is in effect as long as a fluid set point temperature is not exceeded. If the set point is exceeded, the proportional control is internally modified to limit the fluid outlet temperature to the set point as with Type6.
Auxiliary Cooler w/Proportional Control	 Type1246	Type1246 models an external, proportionally controlled fluid cooler. External proportional control (an input signal between 0 and 1) is in effect as long as a lower limit fluid set point temperature is not exceeded. If the lower limit is encountered, the proportional control is internally modified to limit the fluid outlet temperature to the set point.
Pipe/Duct	 Type31	This component models the thermal behavior of fluid flow in a pipe or duct using variable size segments of fluid. Entering fluid shifts the position of existing segments. The mass of the new segment is equal to the flow rate times the simulation timestep. The new segment's temperature is that of the incoming fluid. The outlet of this pipe is a collection of the elements that are pushed out by the inlet flow. This plug-flow model does not consider mixing or conduction between adjacent elements.
Variable Speed Pump	 Type110	Type110 models a variable speed pump that is able to maintain any outlet mass flow rate between zero and a rated value. The mass flow rate of the pump varies linearly with control signal setting. Pump power draw, however, is modelled using a polynomial. Pump starting and stopping characteristics are not modelled, nor are pressure drop effects. As with most pumps and fans in TRNSYS, Type110 takes mass flow rate as an input but ignores the value except in order to perform mass balance checks. Type110 sets the downstream flow rate based on its rated flow rate parameter and the current value of its control signal input.
Heat Exchanger with User-Provided Effectiveness	 Type91b	A zero capacitance sensible heat exchanger is modelled as a constant effectiveness device which is independent of the system configuration. For the constant effectiveness mode, the maximum possible heat transfer is calculated based on the minimum capacity rate fluid and the cold side and hot side fluid inlet temperatures. In this mode the effectiveness is input as a parameter and the concept of an overall heat transfer coefficient for the heat exchanger is not used. In this version, the effectiveness is constant over the timestep as the effectiveness is provided to the model as an input and not a parameter.

<p>2-Pipe Console Unit in Energy Rate Control Mode</p>	 <p>Type673</p>	<p>Type673 models a piece of HVAC equipment commonly known as a two pipe console unit. Such devices pass air across a tube bank that contains either hot or cold fluid. Depending upon the temperature of the air and the fluid, the air will exit either hotter or colder than it entered. Type673 models a two pipe console unit in energy rate control mode, meaning that sensible and latent loads are inputs to the model. Type673 includes a “number of identical units” parameter that allows for easy scaling of the system to meet the building load.</p>
--	--	--

The console unit is the representation of a simple model for the demand side, i.e. the zones, that easily allow the connection between building and equipment simulators in case they are treated in separate ways. The type models a two pipe console unit in energy rate control mode, i.e. the heating and cooling loads (sensible and latent) are inputs to the model, as well as the inlet fluid conditions (temperature) and control signals. The model will calculate the outlet fluid conditions, checking the supplied capacities to make sure the device is able to meet the load. This fan coil accept the simulation of instantaneous systems in coordination with the building model.

The fan coil is chosen instead of a radiator in order to have a more abstract model of the emission to the zone. Thus the handling of physical phenomena is simplified since these effects (e.g. efficiency, transfer rate) are not the main request of the simulation. In other words, it models the interface between the energy distribution system and the zone as a black box where the energy flows pass from one side of the box to the other delivering/capturing energy.

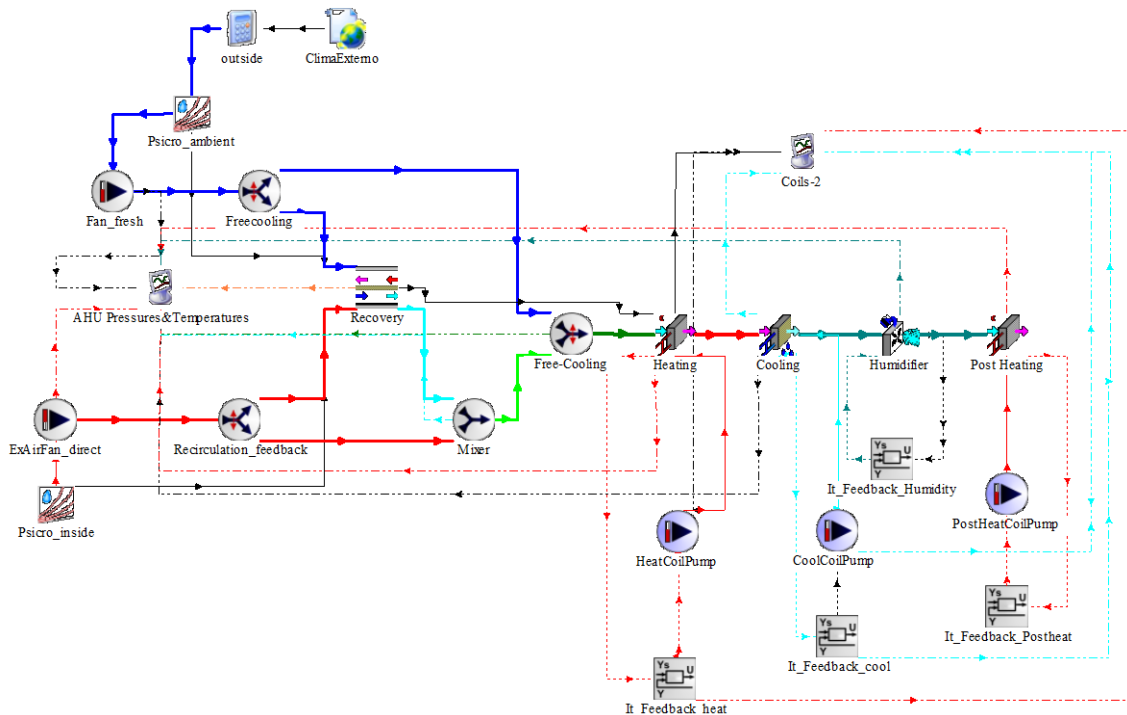
There are three parameter of the console unit that can be directly linked to the IFC data, i.e. rated cooling and heating capacity and the global value of the specific heat of the fluid flow. Two input and output values are linked to the circuit, i.e. inlet and outlet fluid temperature and flow rate. The inputs sensible and latent load are specified within the building model and the heating and cooling control signal are delivered by the APOs within BaaS system. The output parameter "sensible load met", "latent load met" and "load not met" are used for the assessment of thermal comfort. The difference of fluid flow at inlet and outlet of the component is equal to the delivered energy.



**Figure 23: Cross Sectional Schematic of a Two-Pipe Console Unit**

The following figure displays an example of a TRNSYS model of an air handling unit AHU. The main difference while developing a model for air handling units is the transmission medium. But there are several similarities since the transportation of air in a duct system using fans is from the energetic point of view comparable to water pipes and pumps. The air distribution system therefore will base on a similar circuit model, besides some cases where the return flow is missing. Other differences occur in the generation part of the system since air handling units mostly do not have a generation part included like heating and cooling systems, they are usually linked to the external generation systems. Only humidifiers are normally included in the system as well as other components like heat exchangers.

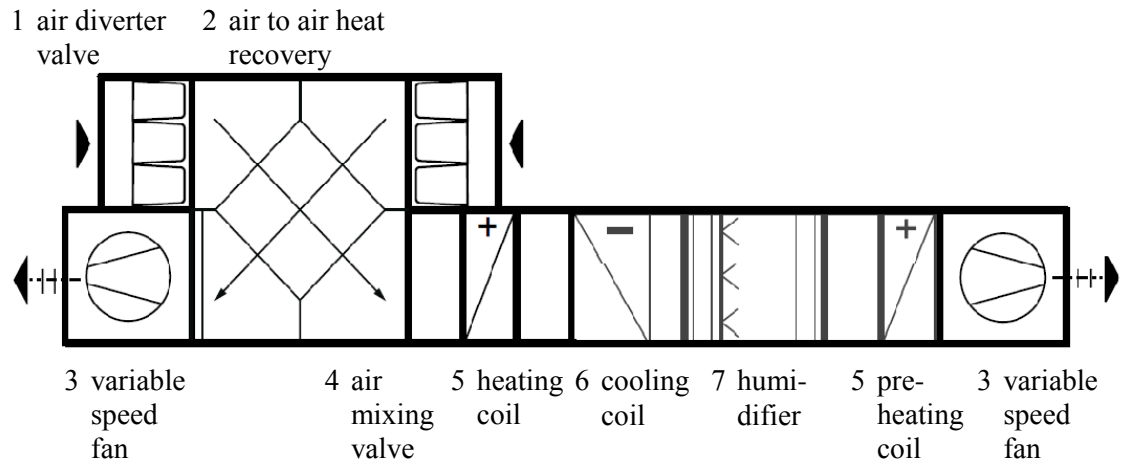
Thus, the developed models for water based systems are also applicable on air handling units, taking into account a different but similar distribution system and including the development of specialised components like humidifier, heat exchanger and heat/cold batteries.



**Figure 24: Example simulation model (TRNSYS) of air handling unit**

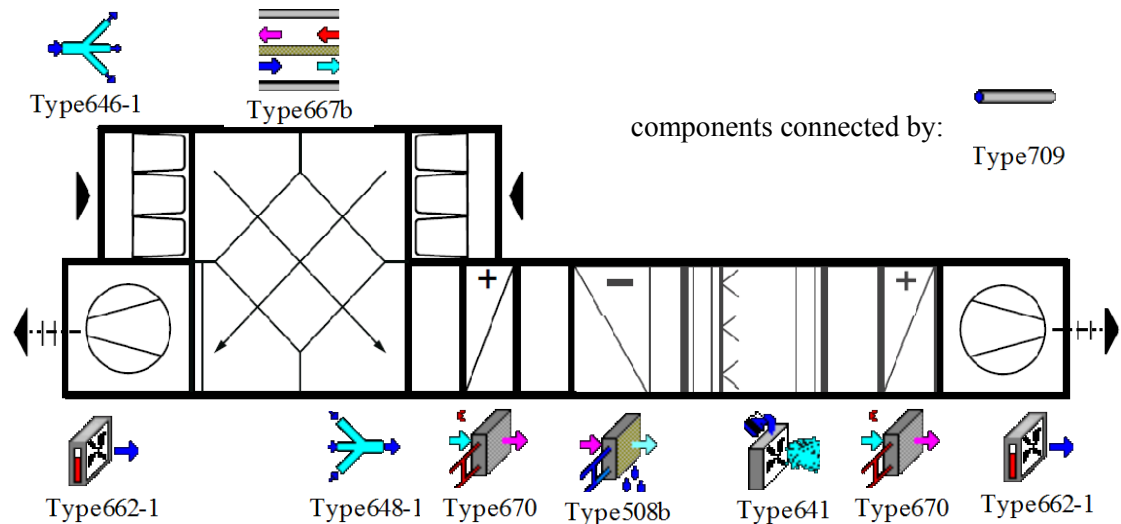
The model for the air handling unit AHU contains several sub-components, thus is more complex compared to heat or cold generation. The following figure displays a technical scheme of an AHU including most of the components that could occur, i.e.

- hydronic equipment: air diverter valve (1), air mixing valve (4), two variable speed fans (3)
- air to air heat recovery (2)
- air treatment: heating and pre-heating coil (5), cooling coil (6), humidifier (7)
- connected by ducts within the unit and with external components, i.e. incoming and outgoing



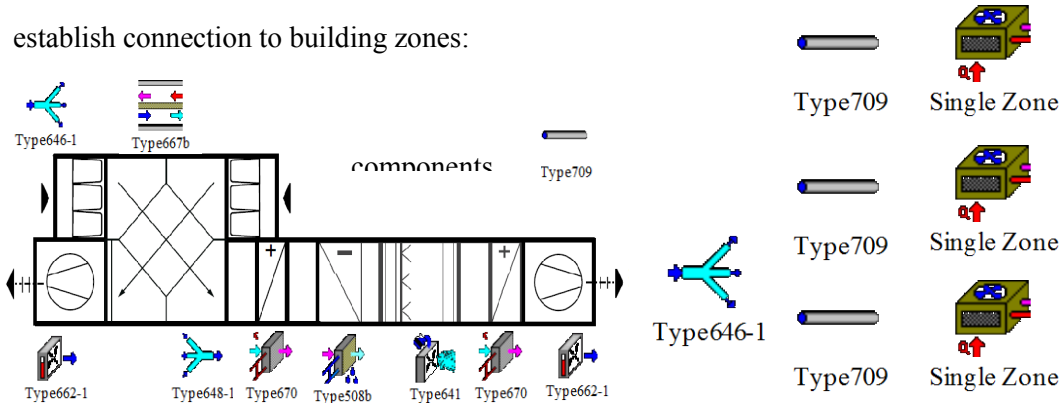
**Figure 25: Air handling unit and sub-components - scheme**

The corresponding TRNSYS types for the simulation model are displayed in the next figure represented by their icon and placed nearby the real technical components.



**Figure 26: Air handling unit and sub-components - scheme and corresponding TRNSYS model (simplified)**




The connection of the entire unit to the building, i.e. 1..n zones, is established by flow diverter, ducts and terminate in different zone models. The TRNSYS type displayed in the following figure acts as a placeholder for any chosen kind of zone model.








**Figure 27: Air handling unit and sub-components - connection to zones**

The following table contains the description of the TRNSYS types used in the sample of the AHU model.

**Table 12: TRNSYS types for components in AHU**

Component	Icon	Description
1 Air diverter valve	 Type646-1	Type646 models a supply air plenum. One inlet flow of air is split up into as many as 100 individual streams whose mass flow rates are user specified fractions of the inlet air flow rate. The limit of 100 inlet flows can be modified in the Fortran source code.
2 Air to Air Heat Recovery	 Type667b Type667d	Type667 uses a “constant effectiveness – minimum capacitance” approach to model an air to air heat recovery device in which two air streams are passed near each other so that both energy and possibly moisture may be transferred between the streams. Because of the “constant effectiveness – minimum capacitance” methodology, the model may be used to model a device with any configuration of air streams (parallel flow, cross flow, counter flow, etc.) and may be used to model the sensible and latent aspects of an air to air heat exchanger, an enthalpy wheel, a hygroscopic heat exchanger or a permeable walled flat plate recuperator, among other devices.
3 Variable Speed Fan/ Blower	 Type662-1	Type662, donated as Type111 to the standard TRNSYS Library with the release of v. 16, models a fan that is able to turn at any speed between 0 (full stop) and its rated speed. While the mass flow rate of air moved by the fan is linearly related to the control signal, the power drawn by the fan at a given flow rate can be any polynomial expression of the control signal. As with most pumps and fans in TRNSYS, Type662 takes mass flow rate as an input but ignores the value except in order to perform mass balance checks. Type662 sets the downstream flow rate based on its rated flow rate



		parameters and the current value of its control signal inputs.
4 Air mixing valve	 Type648-1	Type648 models a return air plenum. Up to 100 individual flows of air are mixed together to determine the properties of the air exiting the plenum. The limit of 100 inlet flows can be modified in the Fortran source code.
5 Heating Coil With Hot-Side Bypass	 Type670	Heating Coil With Hot-Side Bypass to Keep Air-Side Outlet Below its Setpoint  Type670 simulates an air heating coil with an internally controlled bypass damper that acts to maintain the outlet air temperature above the inlet air temperature and below a user-specified set point temperature.
6 Cooling Coil Using Bypass Fraction Approach - Outlet Air Temperature Control	 Type508b	Type508 models a cooling coil using one of four control modes. The cooling coil is modelled using a bypass approach in which the user specifies a fraction of the air stream that bypasses the coil. The remainder of the air stream is assumed to exit the coil at the average temperature of the fluid in the coil and at saturated conditions. The two air streams are remixed after the coil. In its unrestrained (uncontrolled) mode of operation, the coil cools and dehumidifies the air stream as much as possible given the inlet conditions of both the air and the fluid streams. The model is alternatively able to internally bypass fluid around the coil so as to maintain the outlet air dry bulb temperature above a user specified minimum, to maintain the air outlet absolute humidity ratio above a user specified minimum or to maintain the fluid outlet temperature below some user specified maximum.
7 Simple Adiabatic Humidifier	 Type641	This model represents a simple adiabatic humidifier whose outlet air state is determined by an energy balance. Thermal losses from the humidifier are neglected. The model allows for the humidifier not to respond immediately to the control signal but to reach its steady state moisture gain rate exponentially. Furthermore, the model allows the user to determine whether condensate leaves the humidifier at the temperature at which it enters, at the temperature of the air exiting the humidifier or at any point in between.
Circular, Fluid-Filled Pipe/Duct	 Type709	Very much like standard TRNSYS Type31, this component models the thermal behaviour of fluid flow in a pipe or duct using variable size segments of fluid. Entering fluid shifts the position of existing segments. The mass of the new segment is equal to the flow rate multiplied by the simulation time step. The new segment's temperature is that of the incoming fluid. The outlet of this pipe is a collection of the elements that are "pushed" out by the inlet flow. This so-called "plug-flow" model does not consider mixing or conduction between adjacent elements. A maximum of 25 segments are allowed in the pipe. When the maximum is reached, the two adjacent segments with the closest temperatures are combined to

	<p>make a single segment. Where Type709 differs from Type31 is that instead of asking the user to provide an overall UA value for the pipe and its insulation, the user is here asked to provide the physical characteristics of the pipe material, fluid and insulation material.</p>
--	--

### 5.5 Modules for simulation models of subsystems

The following figures contain the generic simulation model for a heating and a cooling circuit, i.e. the system module. The TRNSYS type placed central will print or plot the required simulation output, i.e. KPIs or other values (e.g. efficiencies), depending on kind of subsystem and data request.

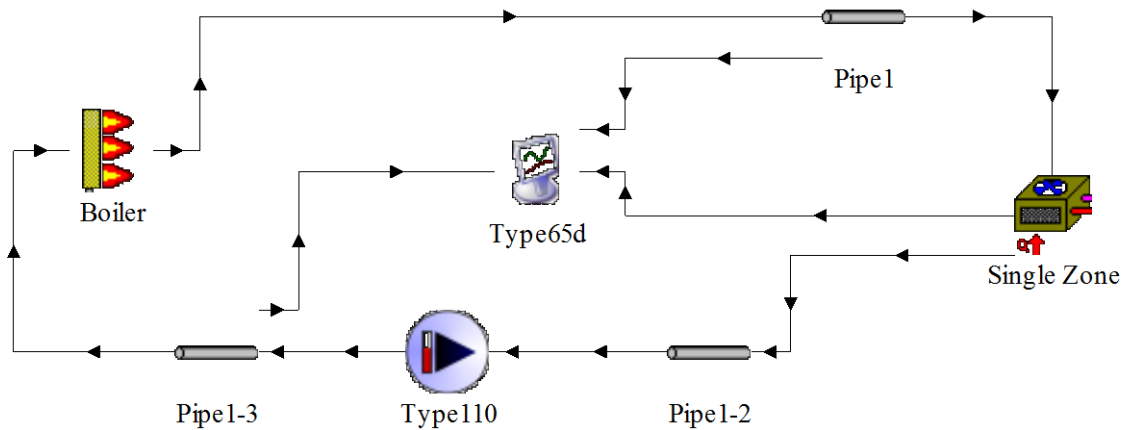


Figure 28: Generic simulation model - heating system module

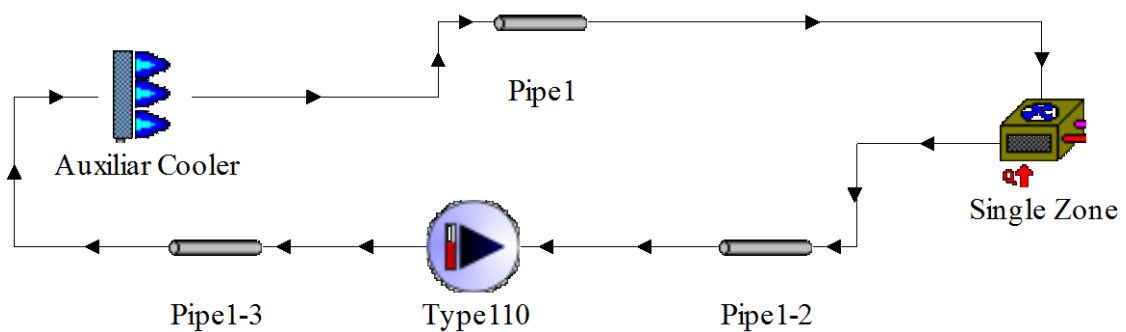


Figure 29: Generic simulation model - cooling system module

### 5.6 Transferability of simulation modules on pilot buildings

Within the BaaS project and WP4 the subtask 4.2.1 focuses the development of thermal simulation models for the pilot buildings in order to estimate the energy performance and thermal comfort behaviour. The described general model for heating and cooling systems as well as air handling units is applicable on all of the systems that will be found in the test-beds and pilot buildings. The following table gives an overview of the buildings regarding thermal simulation models.

**Table 13: Thermal simulation models for the pilot buildings and test-beds**

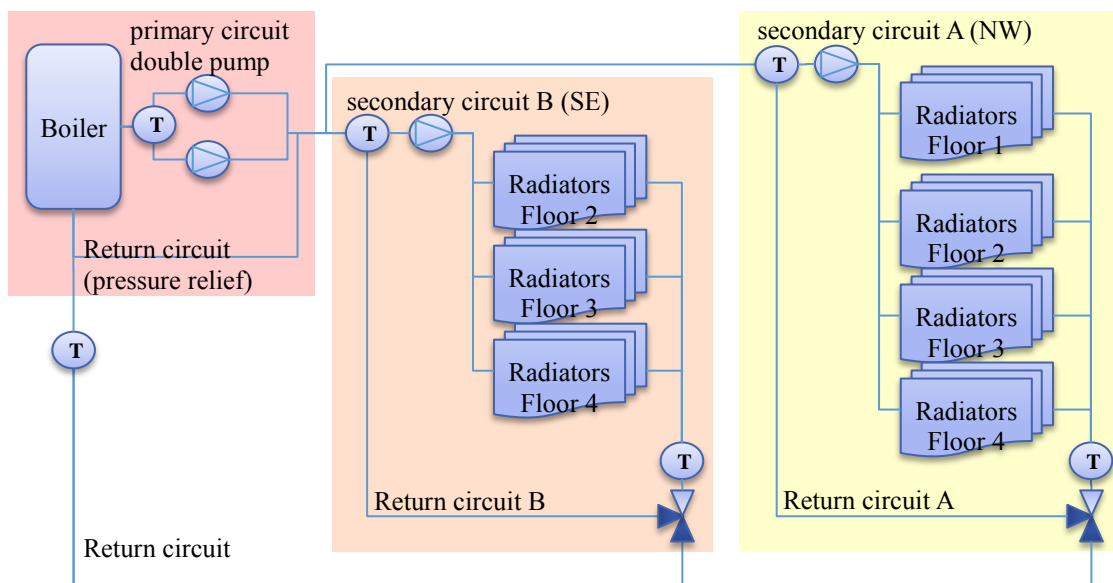
Building	Behaviour	Reaction Time	Solar loads	Occupancy/ Schedule	Systems					
					heating	cooling	AHU	lighting	domestic hot water	
test-bed	TUC, Greece	low inertia	quick	low	working	x	x		x	x
	CARTIF, Spain	low inertia	slow/quick	high	working	x	x	x	x	
pilot	ZUB, Germany	high inertia	slow	high	working	x	x	x	x	
	Hotel, Spain	low inertia	quick	high	booking	x	x	x		x
	School, Spain	low inertia	quick	low	scholar	x				

**5.7 Example for the application of the generic simulation modules on the pilots**

The application of the developed generic simulation models on the pilots will be shown using the school as an example since the heating system installed is relatively simple and fits well the general structure of the modules.

**5.7.1 Abstract simulation model approach**

The heating system and distribution network "in reality", i.e. the "hardware", is displayed in the following figure in a simplified way in order to understand the principle of system and distribution.



**Figure 30: Abstract scheme of heating system**

The three coloured boxes in the back distinguish different sections of the system. These sections will be modelled separately in different "macros", see further explanations below.

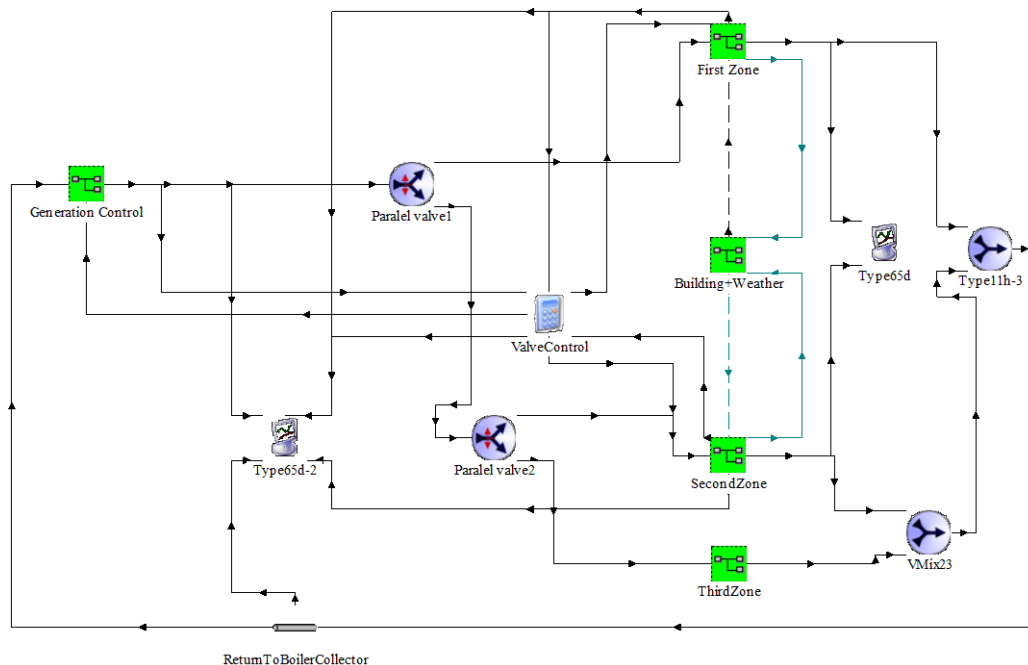
The sketch does not contain building and control as well as all flow diverters and mixers apart from the two valves handling the return circuits of the zones. The model of the school will in future have a third additional circuit for the new building part containing the auditorium. The new circuit will be modelled similar (parallel).

### 5.7.2 Main simulation model

The simulation model for Sierra Elvira School contains - on the main level - several components and macros. In TRNSYS "components" are the main functional objects linked by connections and "macros" are used to group several components into a sub-system, i.e. macros contain a set of components with internal and external connections.

The components on the main level are

- distribution elements (flow diverter and mixer, pipes)
- control function handling (equation, valve control)
- simulation output handling (online plotter for testing and output file generation)
- macros containing sub-systems



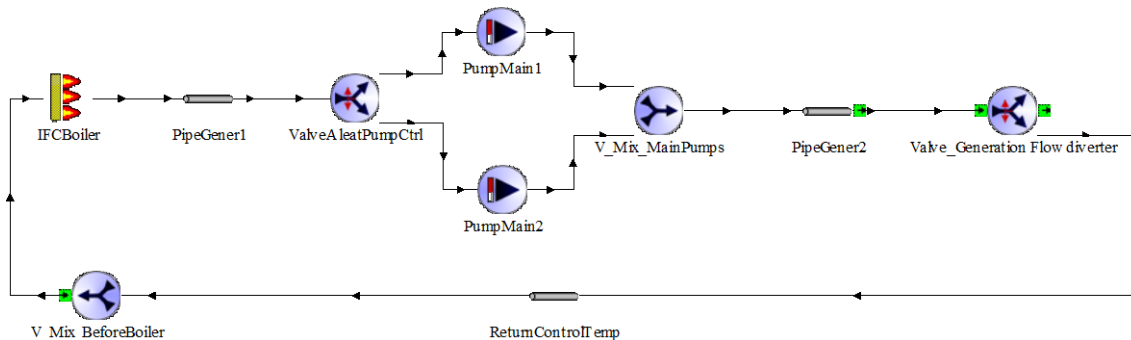
**Figure 31: Entire simulation model of Sierra Elvira School (TRNSYS)**

### 5.7.3 Macro Generation Control

The macro "Generation Control" manages the control of the heat generation and distribution of the primary circuit.

The components contained in macro "Generation Control" are

- heat generator (boiler)
- distribution elements (flow diverter and mixer, pipes)



**Figure 32: Content of macro "Generation Control"**

The components for the pipes are just modelled in detail to handle the physical effect of the delay in the distribution system and the tee pieces just serve as flow mixers. The other relevant components are described in the following subsections.

#### 5.7.4 *Macros zones*

For each zone (in case of simplified zoning model of Sierra Elvira School there will be max. 3 zones) the macro "Zone" enables the access to the control of delivered energy to the zone. The macro is connected to the corresponding physical zone within the building structure (Type56) in order to exchange information from the control side inside the building and process the information about the "reaction" of the physical zone as output of the building.

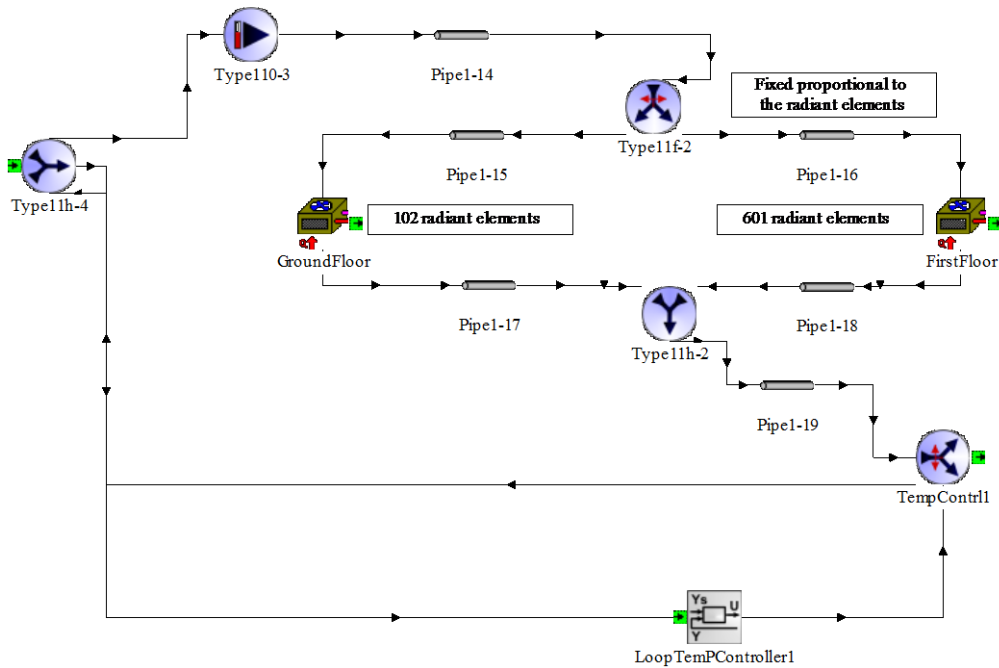
The components contained in macro "FirstZone" are

- radiant elements separated for each floor (fan coil)
- distribution elements (flow diverter and mixer, pipes)
- control function handling (equation, loop temperature control)

The other zones "SecondZone" and "ThirdZone" are similar.

The fan coil is chosen instead of a radiator in order to have a more abstract model of the emission to the zone. Thus the handling of physical phenomena is simplified since these effects (e.g. efficiency, transfer rate) are not the main request of the simulation. In other words, it models the interface between the energy distribution system and the zone as a black box where the energy flows pass from one side of the box to the other delivering/capturing energy.

There are three parameter of the console unit that can be directly linked to the IFC data, i.e. rated cooling and heating capacity and the global value of the specific heat of the fluid flow. Two input and output values are linked to the circuit, i.e. inlet and outlet fluid temperature and flow rate. The inputs sensible and latent load are specified within the building model and the heating and cooling control signal are delivered by the APOs within BaaS system. The output parameter "sensible load met", "latent load met" and "load not met" are used for the assessment of thermal comfort. The difference of fluid flow at inlet and outlet of the component is equal to the delivered energy.



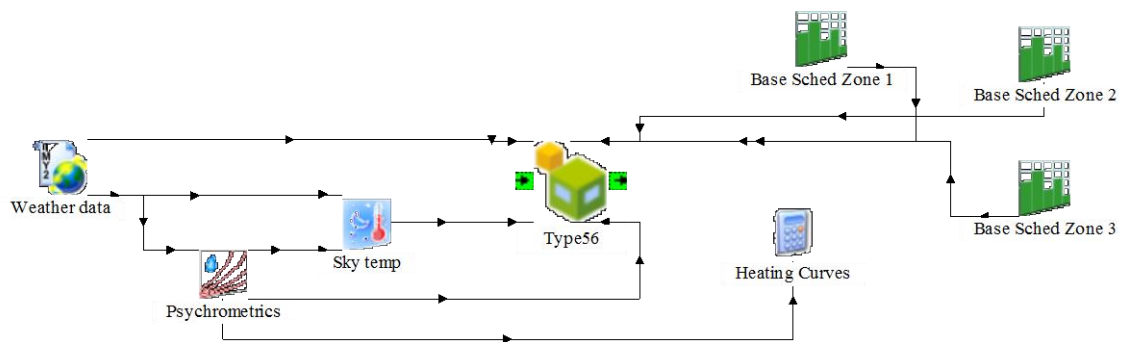
**Figure 33: Content of macro "Macros Zones"**

### 5.7.5 Macro Building + Weather

Within the simulation model the macro "Building+Weather" serves as sub-system to handle the building structure, usage and boundary conditions such as climate data and processing.

The components contained in macro "Building+Weather" are:

- building structure (Type56)
- climate data and processing
- control function handling (equation, heating curves)
- schedules for building use, e.g. occupancy, internal gains



**Figure 34: Content of macro "Building+Weather"**

### 5.8 Model configuration

In addition to the general knowledge about the components included in the building and the simulation model, the relations among them are required as well as the sets of parameter which define the boundary conditions. The next table contains the most relevant parameter for the sample circuit, displayed in groups according to the source.

The first two parameter incoming temperature and incoming flow represent the operational boundaries for the whole circuit and must be available from the underlying building model (IFC). The initial values for capacity, controls etc. are set at starting point of the simulation, i.e.  $t = \tau_0$  and derived from the parameter of the IFC model directly or after certain treatment (processed). During the simulation timesteps ( $t = \tau_n$ ) there are some performance parameter obtained from equations based on physical phenomena, i.e. boiler and pump efficiencies.

Additional parameters are required regarding control signals for the system from APO and data warehouse DW, mainly for the request of actual sensor data or historical data.

Building models can run inside the same TRNSYS simulation tool or running in parallel under different building simulation software, using the same timesteps and external weather conditions (co-simulation). Every conditioned zone must deliver as an output of for the facility simulator, the sensible and latent demand of the zone as well as a Boolean activation of the heating/cooling/AHU system, that will be delivered to the correspondent final elements in charge of giving energy to the zone, and an input connected to the fan coil that represent energy delivered by the fan coil to the zone. As can be seen, the respective output of the building simulator is converted univocally in a fan coil input and vice versa, creating an energy link between facility and building simulators.

This solution also allows testing the behaviour of the heating /cooling/AHU facilities for a determinate listed building demand, or the behaviour of the occupied zone under determinate energy inputs. In both cases, it will be only needed an input/output file representing the no studied/tested component.

**Table 14: Simulation parameter obtained from IFC, data warehouse DWH and BaaS system for generic heating circuit**

System component	Boiler	Pipe	Pump	Emission zone
IFC: model configuration	incoming temperature and flow	incoming temperature and flow	incoming temperature and flow	incoming temperature and flow
IFC: parameter $t = \tau_0$	capacity, min control signal, setpoint temperature		rated flow rate, rated power, motor heat loss fraction, number and values of power coefficients	rated heating and cooling capacity, fluid specific heat
IFC: processed parameter		inside diameter, pipe length,		

$t = \tau_0$		loss coefficient		
IFC: input $t = \tau_n$	boiler efficiency, combustion efficiency		total pump efficiency, motor efficiency	
DWH: parameter $t = \tau_n$		initial temperature		
APO: parameter $t = \tau_n$	control signal, min control signal		control signal	heating and cooling control signal, sensible load, latent load

### 5.9 Mapping the IFC file

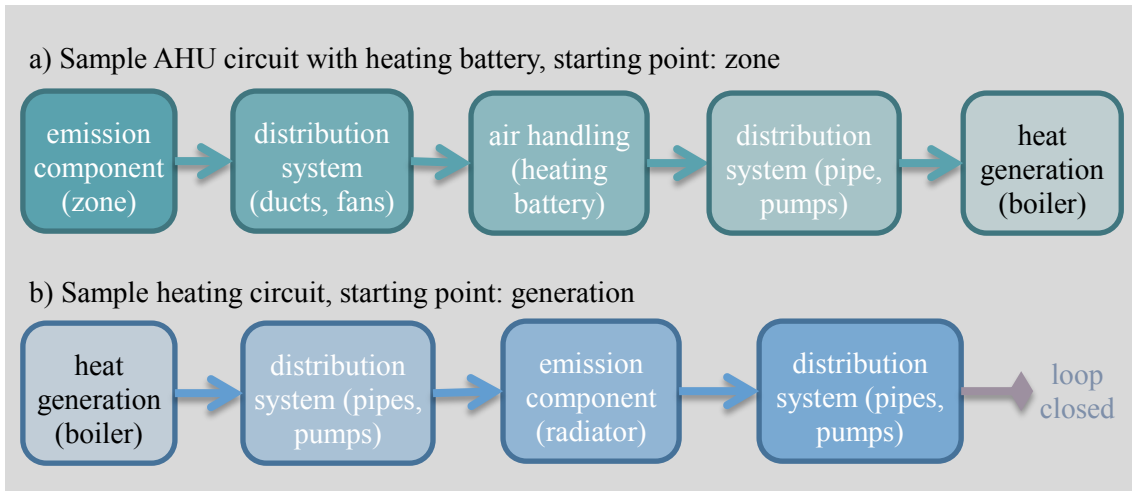
The mapping of the IFC file needs the definition of a concept how to handle the IFC file during the first data access. There are two ways to identify the general structure of the systems contained in the IFC file. For each subsystem (heating, cooling, air handling, lighting, domestic hot water):

- a) either start from the zones/spaces and follow the path from terminal components (e.g. radiators for heating or air outlet for ventilation) to generation or
- b) search for all generating components and find the terminating building demand in the zones/spaces.

The following figure displays exemplarily the two processes using a) an example of an air handling unit (AHU) where emission components (air outlets) are placed in zones and connected via distribution components (ducts and fans) to a component where the "air handling" includes several kinds of treatment e.g. heating, cooling, humidifying, heat exchange. In case of a heating battery the linked distribution system indicates the path to the termination heat generation.

The process vice versa b) is displayed for a simple heating system with boiler as starting point and radiators as indicators in which zone the heat is delivered in order to cover the demand. Following the return flow the return to the boiler closes the loop and adds information to the model that is required for the thermal simulation model, i.e. pump efficiency and temperature level of the return pipe.





**Figure 35: Different principles and samples for the mapping process.**

Regarding several pros and cons the on-going process of the query development will focus on the second presented principle to start from the generation side since even in multiple generation systems the number of starting points will be - in most cases - less than the number of thermal zones within the building model. Additionally the generation will not vary while the structure of zoning can change.

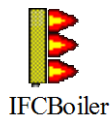
### 5.10 Relation of simulation component parameters and data sources

The analysis of the data correlation and TRNSYS shows that the data have to be treated in several different ways. Some data required for simulation are supported by the IFC, some are internal simulation data without link to IFC and some are somewhere in-between. The data access to other sources (e.g. weather data) will be similar. Additionally some data are constant and some change with time or boundary, i.e. static, dynamic, semi-static or semi-dynamic.

The relation among the data will be explained using the example simulation model of the Sierra Elvira School, introduced in chapter 5.7, and will follow the simulation model components.

The following tables contain the description and correlation of components on parameter level.

#### 5.10.1 Generation of heat and cold (water)



Type 659 models an external, proportionally controlled fluid heater. External proportional control (an input signal between 0 and 1) is in effect as long as a fluid set point temperature is not exceeded. If the set point is exceeded, the proportional control is internally modified to limit the fluid outlet temperature to the set point as with Type 6.



Type 1246 models an external, proportionally controlled fluid cooler. External proportional control (an input signal between 0 and 1) is in effect as long as a lower limit fluid set point temperature is not exceeded. If the lower limit is encountered, the proportional control is internally modified to limit the fluid outlet temperature to the set point.

**The used Types for heat and cold generation are auxiliary heater/chiller with proportional control and represent a simplified physical behaviour. The input and output data are contained in**

Table 15 for the three categories Parameter, Input and Output.

**The static or initial data are derived from the IFC model and marked with “x” or “initial” in the corresponding column in**

Table 15, i.e.

- performance (rated capacity, efficiency, loss coefficient),
- boundary conditions (fluid specific heat) and
- Initial control (control, set-point temperature, fluid temperature and flow).

The dynamic input data from the APO is marked in the next column, i.e.

- mainly control function and set-point temperature and
- In case of relevance - marked “(x)” - additional input signals from the APO could overwrite the static data in first timestep, e.g. the boiler performance data (rated capacity, efficiency) or at every timestep, e.g. the temperature of surroundings (in case of relevance, if boiler heat loss should be simulated in detail, should be sensor data).

Input and output of system status, i.e. fluid temperature and flow rate, are derived from the model configuration. The initial input data for the first timestep is defined static within IFC, for the following timesteps the simulation model will calculate dynamically the system status basing of the knowledge of component links.

**The output parameter to the APO are also marked**

Table 15. The heating rate serves as KPI for indicating the heat production of the boiler. In case of relevance thermal losses and rate of energy could also be requested by the APO.

The required parameter for chiller (cold water generation) are similar to the water boiler.

**Table 15: Parameter, Inputs and Outputs – Waterboiler**

	Name	Value (sample)	Unit	input static IFC	input dyn. APO	model config.	output to APO
Parameter	Rated Capacity	250	kW	x	(x)		
	Fluid specific heat	4.19	kJ/kg.K	x			
Input	Inlet fluid temperature	10	°C	initial		x	
	Inlet fluid flowrate	27000	kg/hr	initial		x	
	Control function	1	-	initial	x		
	Set-point temperature	85	°C	initial	x		
	Overall loss coefficient	0.0	kJ/hr.K	x			
	Boiler Efficiency	0.78	Fraction	x	(x)		
	Temperature of surroundings	20.0	°C		x		
	Output	Outlet fluid temperature	50	°C			x

Outlet flow rate	0	kg/hr	x	(x)
Heating rate	0	kJ/hr		x
Thermal losses	0	kJ/hr		(x)
Rate of energy delivery to fluid stream	0	kJ/hr		(x)

The data required from the IFC model for building up the simulation components will be provided through the requesting APO layer. The correlation of data is actually under construction and can be summarised as follows.

The corresponding IFC4 object for the waterboiler is represented by the IFC entity *IfcBoiler*, where *IfcBoilerTypeEnumeration* is *WATER*. There are several property sets related to this entity, i.e. *IfcBoiler.Pset\_BoilerTypeWater.HeatOutput* contains the required “rated capacity” as input for the TRNSYS component and *IfcBoiler.Pset\_BoilerPHistory.NominalEfficiency* will also be used as input for the simulation model. The inlet and outlet ports are connected as sink and source via the *IfcDistributionPort*. The *Port-Power* and *-Inlet* in IFC assign the energy source, i.e. gas, oil, district heat, biomass (according to the installations in the pilot buildings).

### 5.10.2 Flow diverter



Type11f

The use of pipe or duct tee-pieces, mixers, and diverters which are subject to external control is often necessary in thermal systems. This component has ten modes of operation. Modes 1 through 5 are normally used for fluids with only one important property, such as temperature. Modes 6 through 10 are for fluids, such as moist air, with two important properties, such as temperature and humidity.

This instance of the Type11 model uses mode 2 to model a flow diverter in which a single inlet liquid stream is split according to a user specified valve setting into two liquid outlet streams.

The main function of a flow diverter is:

- a) the allocation of the flow to one of the pumps in the double pump system. Only one pump is exclusively working within time. The control signal switches between 0 and 1 or
- b) the division of the flow delivered to the system and the return circuit.

The physical effects covered with control (b) include

- the handling in case supply temperatures are out of range and
- to ensure that delivered flow rates within the system are matching the demanded flow rates in the secondary circuits in order to keep the system pressure.

The initial value for the control system is defined by default.

The flow diverter just represents a part of the distribution system where only the initial values and the control signal have a high relevance. The output parameter of temperature and flow rate of both outlets could be hand over to the APO in case of relevance but normally the information is only required within the simulation model.

**Table 16: Parameter, Inputs and Outputs - Flow diverter**

	Name	Value (sample)	Unit	input static IFC	input dyn. APO	model config.	output to APO
Parameter	Controlled flow diverter mode	2	-	constant value assigned to diverting valves by TRNSYS			
Input	Inlet fluid temperature	20	°C	initial		x	
	Inlet fluid flowrate	100	kg/hr	initial		x	
	Control signal	0	-	initial	x	(x)	
Output	Temperature at outlet 1	0	°C			x	(x)
	Flow rate at outlet 1	0	kg/hr			x	(x)
	Temperature at outlet 2	0	°C			x	(x)
	Flow rate at outlet 2	0	kg/hr			x	(x)

The corresponding entity in IFC4 is *IfcValve* where *IfcValveTypeEnum* is *DIVERTING*. The initial values for the inlet fluid temperature and flowrate are defined by *Pset\_ValvePHistory.MeasuredFlowRate* and *Pset\_ValvePHistory.PercentageOpen*. The inlet and two outlet ports are connected as sink and source via the *IfcDistributionPort*.

### 5.10.3 Flow mixer



Type11h

A pipe fitting or tee-piece is a junction or transition in a piping flow distribution system used to connect pipe segments, resulting in changes in flow characteristics to the fluid such as direction or flow rate.

This instance of the Type11 model uses mode 1 to model a tee piece in which two inlet liquid streams are mixed together into a single liquid outlet stream.

Flow mixing has the similar structure - compared to flow diverters - with different variables for mixer pattern. No control signal is required for mixers, since they just simply mix the flow. The similarity also include the relevance of possible output parameter to APO, normally they are not used externally.

**Table 17: Parameter, Inputs and Outputs - Flow mixer**

	Name	Value (sample)	Unit	input static IFC	input dyn. APO	model config.	output to APO
Parameter	Controlled flow diverter mode	1	-	constant value assigned to mixing valves by TRNSYS			
Input	Temperature at inlet 1	20	°C	initial		x	
	Flow rate at inlet 1	100	kg/hr	initial		x	
	Temperature at inlet 2	20	°C	initial		x	
	Flow rate at inlet 2	100	kg/hr	initial		x	
Output	Outlet temperature	0	°C			x	(x)

Outlet flow rate	0	kg/hr	x	(x)
------------------	---	-------	---	-----

The corresponding entity in IFC4 is *IfcPipeFitting* where *IfcPipeFittingTypeEnum* is *JUNCTION*. The two inlet and one outlet ports are connected as sink and source via the *IfcDistributionPort*. There is no control signal required.

### 5.10.4 Pump



Type110

A pump is a device which imparts mechanical work on fluids or slurries to move them through a channel or pipeline. A typical use of a pump is to circulate chilled water or heating hot water in a building services distribution system.

The pump parameter concerning performance data, i.e. rated flow rate and rated power, are usually derived from the IFC model. The APO could overwrite the initial values in the first timestep by individual (optimised) values. The other parameters are either kept constant or not used within the simulation model.

The input and output values for fluid temperature and flow rate are - as seen for other components - derived from the model configuration and initialised by IFC data. Main input value for the pump required from APO is the control signal. The efficiencies are - like the performance parameter - also usually derived from IFC or could come from APO.

The only relevant output value to the APO would be the power consumption of the pump. The outlet temperatures and flow rates are normally used internally.

**Table 18: Parameter, Inputs and Outputs – Pump**

	Name	Value (sample)	Unit	input static IFC	input dyn. APO	model config.	output to APO
Parameter	Rated flow rate	7500	kg/hr	x	(x)		
	Fluid specific heat	4.19	kJ/kg.K	x			
	Rated power	2684.0	kJ/hr	x	(x)		
	Motor heat loss fraction	0.0	-	constant = 1			
	Number of power coefficients	1	-	not used			
	Power coefficient	1	kJ/hr	not used			
Input	Inlet fluid temperature	20.0	°C	initial		x	
	Inlet fluid flowrate	0.0	kg/hr	initial		x	
	Control Signal	1	-		x		
	Total pump efficiency	0.6	-	x	(x)		
	Motor efficiency	0.9	-	x	(x)		
Output	Outlet fluid temperature	0	°C			x	(x)
	Outlet flow rate	0	kg/hr			x	(x)
	Power consumption	0	kJ/hr				x

Fluid heat transfer	0	kJ/hr	not used
Environment heat transfer	0	kJ/hr	not used

The IFC4 representation for a pump is *IfcPump*. The rated flowrate is contained in *IfcPump.Pset\_PumpPHistory.Flowrate* and rated power in *IfcPump.Pset\_PumpPHistory.Power*. The same property set also contains the total pump and motor efficiency, i.e. *OverallEfficiency* and *MechanicalEfficiency*.

### 5.10.5 Pipe



A pipe segment is used to typically join two sections of a piping network. A duct segment is used to typically join two sections of duct network.

Type31

The main function of the pipe or duct segments is the interconnection of the system components. Physically they do not have important impact on the system performance, besides the effect of time offset for the fluid flow and heat losses. Nevertheless the IFC model provides geometrical information of inside diameter and pipe length. The loss coefficient will also be derived from the IFC model as well as fluid properties, i.e. specific heat and density.

The parameter initial fluid temperature and the initial input values for temperature and flow rate are - as known for the other components - derived from IFC or provided by the APO. The input value for environmental temperature has to be delivered by APO.

For the output parameter it could be interesting for the APO to get environmental losses or delivered energy, depending on the kind of request.

**Table 19: Parameter, Inputs and Outputs – Pipe**

	Name	Value (sample)	Unit	input static IFC	input dyn. APO	model config.	output to APO
Parameter	Inside diameter	0.4	m	x			
	Pipe length	1.0	m	x			
	Loss coefficient	3.0	kJ/hr.m2.K	x			
	Fluid density	1000.0	kg/m <sup>3</sup>	x			
	Fluid specific heat	4.19	kJ/kg.K	x			
	Initial fluid temperature	10.0	°C		x		
Input	Inlet temperature	10	°C	initial		x	
	Inlet flow rate	100	kg/hr	initial		x	
	Environment temperature	10	°C		x		
Output	Outlet temperature	0	°C			x	(x)
	Outlet flow rate	0	kg/hr			x	(x)

Environment losses	0	kJ/hr	(x)
Energy to pipe/duct	0	kJ/hr	(x)
Change in internal energy	0	kJ	not used
Average temperature	0	°C	not used
Rate of change in internal energy	0	kJ/hr	not used

Pipes in IFC4 are defined as *IfcPipeSegment* or *IfcDuctSegment* in case of systems distributing air instead of water. The geometry is stored in property sets (e.g. *IfcPipeSegment.Pset\_PipeSegmentTypeCommon.InnerDiameter*) or quantity sets (e.g. *IfcDuctSegment.Qto\_PipeSegmentBaseQuantities.Length*)

### 5.10.6 Single zone (radiators)



Type673 models a piece of HVAC equipment commonly known as a two pipe console unit. Such devices pass air across a tube bank that contains either hot or cold fluid. Depending upon the temperature of the air and the fluid, the air will exit either hotter or colder than it entered. Type673 models a two pipe console unit in energy rate control mode, meaning that sensible and latent loads are inputs to the model. Type673 includes a “number of identical units” parameter that allows for easy scaling of the system to meet the building load.

NOTE: This component is intended to take energy rate control loads from a building model and convert them to temperature-level control loads by adding or removing heat from a flow stream.

The terminal element introduced into the simulation to link the building demands and the facilities works as a black box that allow the representation of different IFC types as can be the fan coils, radiators, radiant floors etc. by only defining previously the ratio between convection and radiation heat delivery to the controlled zones.

The significant input values of the components are sensible and latent load since these values enable to adapt the component on different kinds of simulation models and indicate the installed power of the system for the heated/cooled room. The source is marked with two “x<sup>1</sup>” in both columns APO and model configuration to indicate, that the values either come from external (APO) or internal within the simulation model of the building (Type 56 in TRNSYS). That gives us the flexibility to work either with full simulation model or co-simulation with real measurements.

The control signals for heating and cooling as well as fan control are delivered by the APO, normally set to 0 or 1.

Fan power is only used as an auxiliary value for this particular component.

The power supplied to the radiator/zone is represented by the output parameters which indicate the load met (sensible, latent) or not met.

**Table 20: Parameter, Inputs and Outputs - Single zone (radiators)**

	Name	Value (sample)	Unit	input static IFC	input dyn. APO	model config.	output to APO
Parameter	Number of identical units	1	-	not used			
	Rated unit fan power	1000.0	kJ/hr	initial	(x)		
	Rated cooling capacity	0	kJ/hr	initial	(x)		
	Rated heating capacity	64058.4	kJ/hr	initial	(x)		
	Fluid specific heat	4.19	kJ/kg.K	x			
	Number of power coefficients	1	-	not used			
	Power coefficient	0	-	not used			
Input	Inlet temperature	20.0	°C	initial		x	
	TOTAL fluid flow rate	0.0	kg/hr	initial			x
	Sensible load	0.0	kJ/hr		x <sup>1</sup>		x <sup>1</sup>
	Latent load	0.0	kJ/hr		x <sup>1</sup>		x <sup>1</sup>
	Fan control signal	1	-		x		
	Cooling control signal	0	-		x		
	Heating control signal	1	-		x		
Output	Outlet fluid temperature	20	°C			x	(x)
	Outlet flow rate	0	kJ/hr			x	(x)
	Sensible load met	0	kJ/hr				x
	Latent load met	0	kJ/hr				x
	Load not met	0	kJ/hr				x
	Fan power	0	kJ/hr	not used			
	Fan fraction of full-load power	0	-	not used			

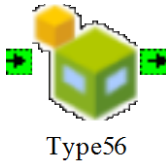
<sup>1)</sup> The sensible and latent load either come from APO or building model (Type 56).

The IFC4 representation for the radiator model is *IfcSpaceHeater*. Space heaters utilize a combination of radiation and/or natural convection using a heating source such as electricity or hot water to heat a limited space or area. Examples of space heaters include radiators, convectors, baseboard and finned-tube heaters. *IfcUnitaryEquipment* could be used for packaged units supporting a combination of heating, cooling, and/or dehumidification; *IfcCoil* could be used for coil-based floor heating.

The related property sets *Pset\_SpaceHeaterPHistory* and *Pset\_SpaceHeaterTypeCommon* contain required attributes like *FractionRadiantHeatTransfer*, *FractionConvectiveHeatTransfer*, *HeatOutputRate*, *OutputCapacity* etc.



### 5.10.7 Building model (Type 56)



This component models the thermal behaviour of a building having multiple thermal zones. The building description is read by this component from a set of external files having the extensions \*.bui, \*.bld, and \*.trn. The files can be generated based on user supplied information by running the pre-processor program called TRNBuild (known as Prebid in TRNSYS versions prior to the release of v. 16.0).

The APO will provide a full data set of climate data depending on the location of the building. The information about the location is stored in the IFC model.

Due to the chosen separation of the building model itself from the energy supply to the radiators (Single zone), the simulation model allow to

- either model the building within the simulation environment using Type 56 in TRNSYS and use the simulation results directly as input to the component “Single zone”
- or use external data as output of co-simulation models provided by the APO and transferred to the building system, i.e. radiators within Single zone component.

The most important output to the surrounding simulation model is the energy demand of all zones since this is the input for the single zone component. For the APO the air temperature of all zones is most relevant. There are several additional simulation parameters thinkable, e.g. comfort parameter, to be transferred to the APO depending on the particular request.

**Table 21: Parameter, Inputs and Outputs - Building model**

	Name	Value (sample)	Unit	input static IFC	input dyn. APO	model config.	output to APO
Parameter	Building geometry, material, zoning <sup>1</sup>		div.	x			
Input	Climate data <sup>2</sup>		div.		x		
	Sensible load met (heating)		kJ/hr		x <sup>3</sup>	x <sup>3</sup>	
	Sensible load met (cooling)		kJ/hr		x <sup>3</sup>	x <sup>3</sup>	
	Latent load met (heating)		kJ/hr		x <sup>3</sup>	x <sup>3</sup>	
	Latent load met (cooling)		kJ/hr		x <sup>3</sup>	x <sup>3</sup>	
	Schedules		-		x <sup>4</sup>	x <sup>4</sup>	
Output	Air temperature of all zones						x
	Sensible energy demand (load) of all zones						x
	Latent energy demand (load) of all zones						x

<sup>1)</sup> Input file (\*.bui or \*.b17) for TRNSYS containing geometry data, material layers and zoning as well as the location of the building.

- 2) Full climate data set from Data Warehouse for specific location of the building, i.e. ambient temperatures, relative humidity, sky/ground temperature, incident and beam radiation for all orientations/inclinations.
- 3) Power (load) either comes from APO or building model (Type 56).
- 4) Schedules for occupancy, sun shading devices, internal gains, operational modes etc. either come from APO or simulation model components located outside of the building model (Type 65).

The representation of the building within TRNSYS is a external input file (\*.bui or \*.b17) containing geometry data, material layers and zoning as well as the location of the building (in order to assign the corresponding climate data).

Usually in the TRNSYS Simulation Studio environment the tool to be used to edit the building file is TRNBuild. The building model bases on the definition of building zones (or air nodes) with a complete data set of the whole thermal envelope (walls, floors, roofs, windows) forming the geometry model. The Version 17 of TRNSYS provides a detailed radiation mode (radiation exchange of surfaces) which has also be supplied be sufficient geometrical data input.

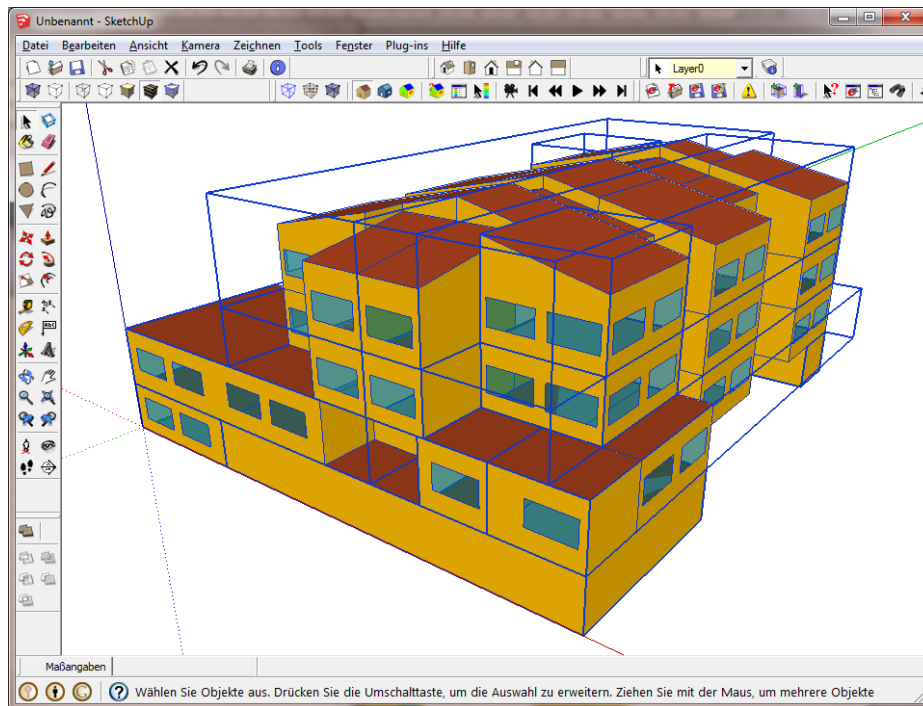
Assigned to the envelope the constructions (material layers, thermal behaviour, window properties, etc.) and shading devices are defined.

The building input file also contains - among other data regarding simulation control and output - usually the definition of

- infiltration and ventilation types and schedules,
- internal gain and controllable device types and schedules and
- heating, cooling and humidity regime data and control functions.

In case of the BaaS simulation models especially the control functions and schedules are placed out of the building model within the surrounding simulation model of the whole system. This is an alternative approach in order to provide direct access of the APO layer to the controllable components. Thus, the mentioned type definitions, schedules and control functions within the building model (Type 56) will stay in our case empty or filled with default values, controlled from "outside".

The simulation tools TRNSYS and Energy Plus provide - among others - with a plugin for the geometrical tool SketchUp the possibility to design the geometrical shape of the building with a 3D-GUI. This allows to generate a complete data set of geometrical data including required geometrical references for the detailed radiation mode (see description above for TRNSYS17). Additionally the assignment of material is supported by the tool as well as the definition of zones. The result of the plugin is represented by an exported file called IDF. In case of TRNSYS the building editor TRNBuild provides the possibility to import this file directly and transform it into the required building input file, i.e. bui or b17. Figure 36 contains an example for the SkechUp model, i.e. the Sierra Elvira School.



**Figure 36: Screenshot Sketch up model of school (IDF file)**

Currently there is no tool available to extract this geometrical data required for the simulation model directly from the IFC-file. With the possibility of creating an intermediate step (in between IFC and simulation model) using the SketchUp plugin and creating an IDF-file, we can - currently - fill the gap. One possibility for the future would be the availability of a "translator" which automatically transforms IFC into IDF files, or to develop own methods for direct data access to the geometrical data of the IFC model, using the the Space boundaries described in section 3.2.1 or Model View Definitions MVD described in section 4.

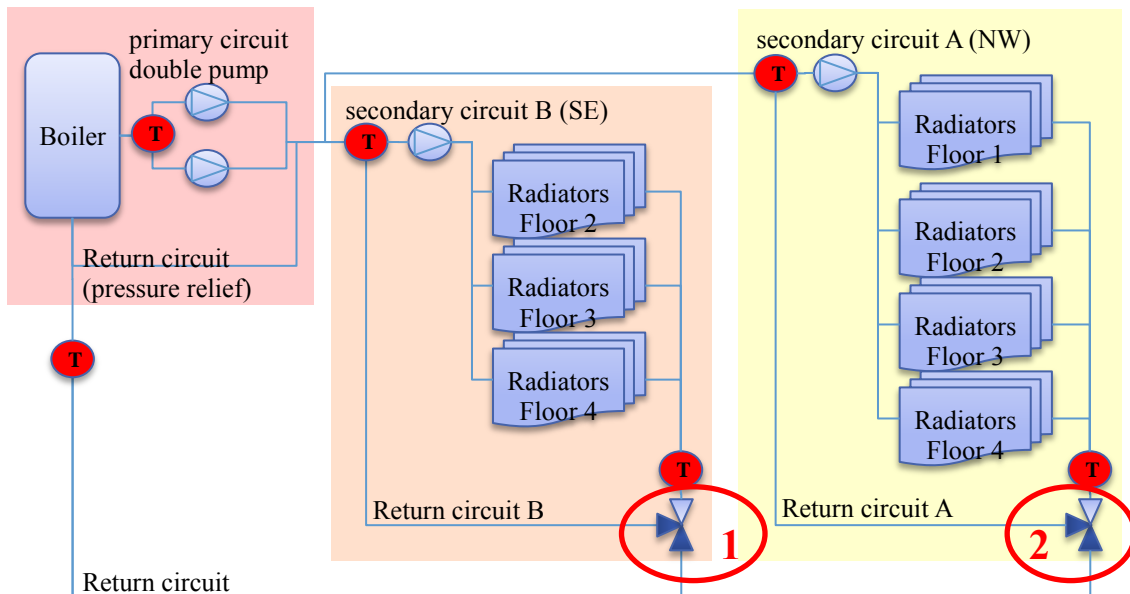
### 5.10.8 *Co-simulation configuration*

The background of this example is to show how this theoretical model will be applied on our example, i.e. the school and the use case to optimize the night setback of the heating system.

#### 5.10.8.1 *Control signal and flow temperatures*

The control signal provided as output of the optimization routine in case of this example is the control signal of the valves - marked red with "1" and "2" in Figure 37.

The realisation of the night setback within the heating system is the control of the amount of water delivered to the return circuit for both secondary circuits A and B. If the system should decrease the temperature level during night, the valve reduces the amount of water delivered back to the boiler. Thus, less new water (heated up to the setpoint) from the boiler will be supplied to the secondary circuits. The mixing of new heated water and the remaining colder water in the return reduces the temperature level in the whole circuit. So if the flow of the supply is decreasing, the resulting temperature if the mix is decreasing.



**Figure 37: Abstract scheme of heating system - interconnection to control optimisation**

The control signal input for the simulation component representing the valve is a value between 0 and 1 and defines the amount of flow delivered to each outlet, i.e. internal return to circuit or external return to boiler. The controller producing the control signals utilizes temperature measurements from different sensors/points of the flow system. The co-simulation configuration allows the controller calculations to either be part of the simulation model or performed using a controller APO module (outside of the simulation). In the latter case, the simulated sensor measurements will be part of the simulation outputs, along with the cost function and constraints values.

In this direction, the management of temperature reduction due to flow mixing can either use the information of:

- the supply temperatures at boiler outlet and the two inlets of the secondary circuits; or
- the return temperatures placed before the valves of the secondary circuits and at the inlet of the boiler.

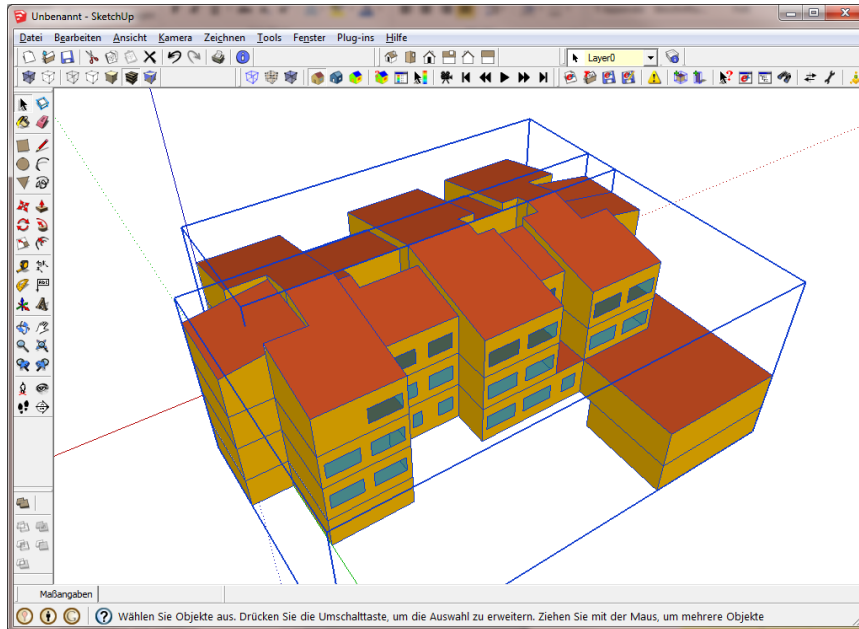
The placement of the temperature is marked with red circles filled with "T" in Figure 37.

### 5.10.8.2 Simulation control

The general control data for the simulation run contains: a) the choice of simulation model type suitable for the optimization task and the required zoning model; b) the definition of the simulation outputs; and c) the general simulation control variables. For the example presented here, the following selections are made:

- A simplified model with 2 zones is sufficient since the overall energy demand is requested, i.e. a model where the zoning is dividing the building regarding the two secondary circuits. Figure 38 shows a 3D-View of the simulation model as screenshot in SketchUp;
- The definition of simulation output request for the example depends on the needs of cost function input and constraints input for the optimization procedure. Of course, more outputs (e.g. control design-relevant temperatures) can be requested;

- c) The general simulation control variables include simulation start and stop time, simulation "warming up period" and simulation time step. In our example the warming up could be a few days depending on the building thermal mass (reaction time), the whole period (start and stop) surrounds one day and the time step should not be too detailed, i.e. hourly simulation should be sufficient.



**Figure 38: Screenshot SketchUp model of school - two zones**

### 5.10.8.3 *Simulation output to APO*

If we define as the goal of the optimization process the reduction of the energy consumption, then the first output of the model should be the boiler consumption. For the comfort constraints, we only need to validate the simulated room temperatures to the reference setpoint temperatures. Note here, that in this case the simulation model can estimate if the demand can be covered by the specific system and on the current boundary conditions, as an alternative index of comfort satisfaction.

### 5.10.8.4 *Sensor data and sensor mapping*

Both the heat generator and the pipes of the distribution system need as an input the information about the temperature of surroundings. These input data are not of high relevance for the estimation of the thermal system behaviour but in case of availability the simulation model could take into account the physical effects of the ambient temperatures of the rooms where boiler and pipes are installed. In order to define clearly the location of sensors and other components, the BaaS system will use the Global Unique ID GUID derived from the IFC model.

### 5.10.8.5 *Schedules*

For the simulation model, schedules are required containing general information about e.g. occupancy and mainly for all controllable devices - at least as initial values to be overwritten by

optimized control signal provided by the APO. This could be heating system schedules as well as schedules for operation of shading devices. In our example, schedules for occupancy should be sufficient in order to have e.g. a default definition of internal gains.

**5.10.8.6 Climate data and weather forecast**

The simulation requires a full climate dataset for the specific location of the building, i.e. ambient temperature, relative humidity, sky/ground temperature, incident and beam radiation for all orientations/inclinations. Normally these data represent average climate behavior (locally) for the requested time period (e.g. one average week end of January) but do not represent the present situation. Additionally, in case of online optimization, data from weather forecast services can be used in order to represent the future climate conditions. In the latter case, dedicated modules are responsible for calculating missing data points (e.g. diffuse and direct solar radiation), based on existing data points (e.g. cloud coverage index).

**5.10.8.7 Building model and HVAC system**

The building model will be derived from the BIM. The geometrical model of the building (envelope) is currently not derived directly from the IFC, instead for TRNSYS we use an in-between handmade IDF model (generated with SketchUp), since there is actually no tool available to allow direct access to the IFC geometry. The IDF model provides information about the properties of the building envelope, i.e. material layers, window data and shading devices.

The building HVAC system is contained in the IFC file and will be mapped as described in D4.1 in order to build up the "topology" of the simulation model and the individual simulation components. The knowledge about the distribution system composition (relation between components) allows to semi-automatically generating the entire simulation model according to the developed simulation modules of D4.1.

**For single data exchange values for certain simulation model components please refer to the end of**

Table 15.

**5.10.8.8 Summary of data exchange for the example**

The table summarizes the aforementioned single items of required data exchange for the presented example. The table could be read along with the corresponding section 4.10 in deliverable D4.1 where the 'Relation of simulation component parameters and data sources' is explained in detail.

**Table 22: Summary of data exchange for the example**

Data exchange	Addressed simulation model component	input static IFC	input from APO	model configuration	output to APO
Control signal	valves in return circuits (flow diverter)		x		
Flow temperatures	distribution system				x

Simulation control	boundary of simulation model and run			x	
Simulation output to APO	energy delivered to the heat generator				x
	building model, resulting room temperatures of zones				x
	representation of radiators: "Load not met"				x
Sensor data	boiler and pipes: temperature of surroundings		x		
	sensor mapping, i.e. IFC GUI vs. sensor naming in BMS			x	
Schedules	building model, occupancy for internal gains		x		
Climate data and weather forecast	weather data boundary		x		
Building model	building geometry, HVAC system, parameter	(x) IDF			
HVAC system	mapping of IFC: system composition, components of simulation model	x			
	general: flow medium properties (specific heat)	x			
	general: initial values for all required dynamic parameter			x	
	boiler: rated capacity, loss coefficient, efficiency	x			
	pump: flow rate, power, efficiency	x			
	pipe: geometry, properties				
	representation of radiators: power, capacity	x			

The analysis of the table summarizing the data exchange against the corresponding tables in D4.1 section 4.10 shows that all required interconnections are solved. Besides two inputs for the radiator representation, i.e. heating and cooling control signal which should be set to 0 in case the schedule of heating or cooling operation is shut down or 1 in case of operation mode.

## 6 Requirements for thermal simulation models for demo sites

In order to address the defined use cases for BaaS (see D5.1 – Functional and interoperability requirements for building services), the thermal simulation models developed should be able to work under the inputs needed (as controllable inputs) and offer as results the constraints and the variables of the cost functions defined for optimizing the building performance. The following table summarizes those inputs and outputs needed for each of the use cases defined for this purpose. Under these premises, D4.2 will develop the necessary simulation models, considering the tools and calculation methodologies analysed in this document and also the BaaS requirements defined in section 3, regarding the simulation functional requirements.

The equations related to the controllable inputs and the model outputs that are summarized in this table can be found in D5.1.

**Table 23: Prediction model requirements**

Input/output	Use Cases	Description
<b>TUC BUILDING MODEL</b>		
<b>Control Inputs</b>		
<b>AC: Mode</b>	TUC_Uc1, TUC_Uc2	Regulate Heating/Cooling mode of the AC system in each office
<b>AC: On/Off</b>	TUC_Uc1, TUC_Uc2	Operate or shut-down the AC system in each office
<b>AC: Setpoint</b>	TUC_Uc1, TUC_Uc2	Regulate the AC set point in each office
<b>Valves: On/Off</b>	TUC_Uc2	Open or close each of the eight thermoelectric valves, regulating the hot water flow in each of the eight branches
<b>Model Outputs</b>		
<b>Ambient Temperature</b>	TUC_Uc1, TUC_Uc2	Office air temperature used in the calculation of the different comfort constraints
<b>Radiant Temperature</b>	TUC_Uc1, TUC_Uc2	Office radiant temperature used in the calculation of the different comfort constraints
<b>Relative Humidity</b>	TUC_Uc1, TUC_Uc2	Office relative humidity used in the calculation of the different comfort constraints
<b>Occupancy</b>	TUC_Uc1, TUC_Uc2	Office occupancy used in the calculation of the different comfort constraints



<b>Total Electric Consumption</b>	TUC_Uc1, TUC_Uc2	Total electric consumption of the building, used for the calculation of the cost function
<b>Total Thermal Energy Consumption</b>	TUC_Uc2	Total thermal energy consumption of the building, used for the calculation of the cost function
<b>Fanger PPD index</b>	TUC_Uc1, TUC_Uc2	Calculation of Fanger PPD index for each office directly from the model
<b>ASHRAE polygon violations</b>	TUC_Uc1, TUC_Uc2	Calculation of ASHRAE polygon violations for each office directly from the model
<b>CARTIF BUILDING MODEL</b>		
<b>Control Inputs</b>		
<b>Radiant floor valves</b>	CAR_Uc1a, CAR_Uc1c, CAR_Uc1e, CAR_Uc1f	Regulate operation (Open/Close) of radiant floor valves for each zone
<b>Setpoint offset of WSHP units</b>	CAR_Uc1b, CAR_Uc1c, CAR_Uc1d, CAR_Uc1f	Regulate set point offset in each zone
<b>Return temperature setpoint of cooling tower</b>	CAR_Uc1d, CAR_Uc1f	Regulate the return temperature set points used as input to the PID controller of the Cooling Tower
<b>Supply water temperature setpoint in TAS</b>	CAR_Uc1e, CAR_Uc1f	Regulate the supply temperature set points used as input to the PID controller of the TAS distribution circuit (valve V3V.8)
<b>Supply temperature setpoint of WSHP distribution circuit (inertia tank)</b>	CAR_Uc1f	Regulate the supply temperature set point used as input to the PID controller of the valve heating the inertia tank
<b>Model Outputs</b>		
<b>Ambient Temperature</b>	CAR_Uc1	Office air temperature used in the calculation of the different comfort constraints
<b>Total Thermal Energy Consumption of TAS distribution circuit</b>	CAR_Uc1a, CAR_Uc1c, CAR_Uc1e, CAR_Uc1f	Total thermal energy consumption of all TAS branches, used for the calculation of the cost function
<b>Total Operation time of WSHP units</b>	CAR_Uc1b, CAR_Uc1f	The sum of the time periods in which the WSHP units are working, used for the calculation of the cost function

<b>Total Electric Consumption of WSHP units</b>	CAR_Uc1c, CAR_Uc1d	Total Electric consumption of WSHP units, used for the calculation of the cost function
<b>Total Thermal Energy Consumption of WSHP distribution circuit</b>	CAR_Uc1c, CAR_Uc1f	Total thermal energy consumption of WSHP distribution circuit, used for the calculation of the cost function
<b>Electric consumption of Cooling Tower</b>	CAR_Uc1d	Total electric consumption of the cooling tower, used for the calculation of the cost function
<b>ZUB BUILDING MODEL</b>		
<b>Control Inputs</b>		
<b>Blinds: Angle</b>	ZUB_Uc1, ZUB_Uc2	Regulate the angle of the blinds in each office
<b>Blinds: Position</b>	ZUB_Uc1, ZUB_Uc2	Regulate the position of the blinds in each office
<b>Floor Slabs: On/Off</b>	ZUB_Uc1, ZUB_Uc2	Open or close the water supply to the floor slabs in each office
<b>Ceiling Slabs: On/Off</b>	ZUB_Uc1, ZUB_Uc2	Open or close the water supply to the ceiling slabs in each office
<b>Hot Water Setpoint</b>	ZUB_Uc1, ZUB_Uc2	Regulate the water temperature set point that supplies the building slabs
<b>Model Outputs</b>		
<b>Ambient Temperature</b>	ZUB_Uc1, ZUB_Uc2	Office air temperature used in the calculation of the comfort constraint and model-based comfort assessment
<b>Operative Temperature</b>	ZUB_Uc1, ZUB_Uc2	Office operative temperature used in the calculation of the comfort constraint and model-based comfort assessment
<b>Relative Humidity</b>	ZUB_Uc1, ZUB_Uc2	Office relative humidity used in the calculation of the comfort constraint and model-based comfort assessment
<b>Occupancy</b>	ZUB_Uc1, ZUB_Uc2	Office occupancy used in the calculation of the comfort constraint and model-based comfort assessment
<b>Luminance</b>	ZUB_Uc1, ZUB_Uc2	Office luminance levels used in the calculation of the comfort constraint and model-based comfort assessment

CO <sub>2</sub>	ZUB_Uc3	CO <sub>2</sub> levels in lecture hall
<b>Total Electric Consumption</b>	ZUB_Uc1	Total electric consumption of the building, used for the calculation of the cost function
<b>Total Thermal Energy Consumption</b>	TUC_Uc1	Total thermal energy consumption of the building, used for the calculation of the cost function
<b>HUSA BUILDING MODEL</b>		
<b>Control Inputs</b>		
<b>Season Change</b>	HUSA_UC1	Determine if the system will operate in summer or winter model (Heating/Cooling)
<b>Hot Water Setpoint Temperature</b>	HUSA_UC1	Determine the hot water temperature set point
<b>Heating/Cooling System availability</b>	HUSA_UC1	Regulate the heating/cooling schedule
<b>Model Outputs</b>		
<b>Ambient Temperature</b>	HUSA_UC1	Air temperature in selected corridors and rooms for calculation of comfort constraint
<b>Total Electric Consumption</b>	HUSA_UC1	Total electric energy consumption measured in generation chiller, used for the calculation of the cost function in cooling mode
<b>Total Thermal Energy Consumption</b>	HUSA_UC1	Total thermal energy consumption measured in the primary generation boilers, used for the calculation of the cost function in heating mode
<b>SIERRA ELVIRA BUILDING MODEL</b>		
<b>Control Inputs</b>		
<b>Heating Schedule</b>	SE_Uc1	Regulate the heating schedule
<b>Heating Setpoints</b>	SE_Uc1	Regulate the supply temperature set point in each zone
<b>Model Outputs</b>		
<b>Ambient Temperature</b>	SE_Uc1	Zone air temperature used in the calculation of the comfort constraint
<b>Total Thermal Energy Consumption</b>	SE_Uc1	Total thermal energy consumption measured in the primary generation boiler, used for the calculation of the cost function

## 7 References

- [1] CARTIF and DALKIA. BaaS Deliverable 1.1, Appendix C: Summary of Problem and Activity Requirements. September 2012.
- [2] CEN. EN15251: Indoor environmental input parameters for design and assessment of energy performance of buildings - addressing indoor air quality, thermal environment, lighting and acoustics. Brussels, 2007.
- [3] D.B. Crawley et al. EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings*, 2001: 319--331.
- [4] S.A. Klein, W.A. Beckman, and J.A. Duffie. TRNSYS - A TRANSIENT SIMULATION PROGRAM. *ASHRAE Transactions* 82 (1976): 623-633.
- [5] M. F. Pichler, A. Dröscher, H. Schranzhofer, A. Contantin, Max Huber, G. Giannakis, N. Exizidou, and D.V. Rovas. Simulation Model Improvements. PEBBLE Deliverable 2.3, 2011.
- [6] PEBBLE. "Positive Energy Buildings thru Better control dEcisions." 2010.
- [7] S. Privara, Z. Vana, D. Gyalistras, J. Cigler, C. Sagerschnig, M. Morari and L. Ferkl. Modelling and identification of a large multi-zone office building. *Control Applications (CCA), 2011 IEEE International Conference on*. IEEE, 2011. 55-60.
- [8] S. Privara, J. Cigler, Z. Vana et. al. Building modelling as a crucial part for building predictive control. *Energy and Buildings*, 2012
- [9] D.B. Crawley, J.W. Hand, M. Kummert, B.T. Griffith. Contrasting the Capabilities of Building Energy Performance Simulation Programs. *Proceedings of Building Simulation 2005, Montreal, Quebec, Canada, IBPSA*, 231-238 pp.
- [10] M. Weise, T. Liebich, R. See, V. Bazjanac, T. Laine and B. Welle. Implementation guide: Space boundaries for energy analysis. *US General Services Administration (GSA) and Open Geospatial Consortium (OGC)*, 2011.
- [11] M. Yalcintas and S. Akkurt. Artificial neural networks applications in building energy predictions and a case study for tropical climates. *International journal of energy research*, (29), pp. 891-901, 2005.
- [12] ASHRAE. *Psychrometrics*. ASHRAE Handbook - Fundamentals (SI), 1:pp.1-16, 2009.
- [13] ASHRAE. *Ventilation and infiltration*. ASHRAE Handbook, 16: pp.1-36, 2009.
- [14] K. Moss. *Heat and mass transfer in buildings*, Taylor & Francis, Jul 16, 2007.
- [15] G.N. Lilis, G.I. Giannakis, D.V. Rovas and E.B. Kosmatopoulos. Energy efficient building ventilation control using the finite element modelling of SRC. *IEEE European Modelling Symposium*, Malta 14-16 Nov. 2012.
- [16] International Organization for Standardization. ISO 13790: 2008: Thermal performance of buildings-Calculation of energy use for space heating and cooling. Geneva, Switzerland: International Organization for Standardization, 2008
- [17] EPBD. Implementing the Energy Performance of Buildings Directive. Brussels, 2011. url: [http://www.epbd-ca.org/Medias/Downloads/CA\\_Book\\_Implementing\\_the\\_EPBD\\_Featuring\\_Country\\_Reports\\_2010.pdf](http://www.epbd-ca.org/Medias/Downloads/CA_Book_Implementing_the_EPBD_Featuring_Country_Reports_2010.pdf)

- 
- [18] V. DIN. DIN 18599-1. Energetische Bewertung von Gebäuden--Berechnung des Nutz-, End, 2007
- [19] V. Bazjanac. Space boundary requirements for modelling of building geometry for energy and other performance simulation. CIB W78, 2010.
- [20] V. Bazjanac and A. Kiviniemi. REDUCTION, SIMPLIFICATION, TRANSLATION AND INTERPRETATION IN THE EXCHANGE OF MODEL DATA. W78 Conference proceedings, pp. 163-168, Maribor, Slovenia, 2007.
- [21] S. Privara, Z. Vana, E. Zavcekova and J. Cigler. Building Modelling: Selection of the Most Appropriate Model for Predictive Control. Energy and Buildings, 2012 (to appear)
- [22] S. Privara, Z. Vana, J. Cigler. and L. Ferkl. Predictive control oriented subspace identification based on building energy simulation tools. 20th Mediterranean Conference on Control & Automation (MED), pp. 1290–1295, 2012.
- [23] S. Privara, J. Siroky, L. Ferkl, J. Cigler. Model predictive control of a building heating system: The first experience. Energy and Buildings, 43:2, pp. 564572, 2011.
- [24] D. Kolokotsa, A. Pouliezos, G. Stavrakakis and C. Lazos. Predictive control techniques for energy and indoor environmental quality management in buildings. Building and Environment, 44:9, pp. 1850–1863, 2009.
- [25] K. Deng, P. Barooah, P.G. Mehta and S.P. Meyn. Building thermal model reduction via aggregation of states. American Control Conference, pp. 51185123, 2010.
- [26] EN ISO 13791: Building information modelling – Information delivery manual – Part 1: Methodology and format. Calculation of Internal Temperatures of a Room in Summer without Mechanical Cooling – General Criteria and Validation Procedures
- [27] ANSI/ASHRAE Standard 140-2001: Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs.
- [28] Häfele, Liebich: IFC Implementation Agreement - Space Boundary Overview on the common agreements for implementing space boundaries. 2009. <http://www.buildingsmart-tech.org/downloads/accompanying-documents/agreements/IFC2x3-space-boundary-implSummary-2010-03-22.pdf/view>.
- [29] Weise, Liebich, See, Bazjanac, Laine: IFC Implementation Guide - Space boundaries for thermal analysis. 2009. <http://www.buildingsmart-tech.org/downloads/accompanying-documents/agreements/IFC2x3-space-boundary-implAgreement-2009-09-17.pdf/view>.
- [30] ISO 16739 (2013). “Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries”. The International Organization for Standardization.
- [31] M. Wetter. Building Controls Virtual Test Bed. Lawrence Berkeley National Laboratory, 2011.
- [32] G. Fraisse, C. Viardot, O. Lafabrie and G. Achard. Development of a simplified and accurate building model based on electrical analogy. Energy and buildings, 34(10):1017, pp.1031, 2002.

- 
- [33] R. Perez, P. Ineichen, R. Seals, J. Michalsky and R. Stewart. Modelling daylight availability and irradiance components from direct and global irradiance. *SolarEnergy*, 44(5), pp. 271-289, 1990.
- [34] J. Hofierka and M. Suri. The solar radiation model for open source gis: implementation and applications. International GRASS users conference in Trento, Italy, 2002.
- [35] K. Moss. Heat and mass transfer in buildings, Taylor & Francis, Jul 16, 2007.
- [36] P. Stefanizzi, A. Wilson and A. Pinney. Internal long-wave radiation exchange in buildings: Comparison of calculation methods: II testing of algorithms. *Building Services Engineering Research and Technology*, 11(3): pp. 87-96, 1990.
- [37] B. Dong. A comparative study of the ifc and gbxml informational infrastructures for data exchange in computational design support environments. *Proceedings of Building Simulation*, (1), pp. 1530-1537, 2007.
- [38] G.N. Lilis, G.I. Giannakis, D.V. Rovas and E.B. Kosmatopoulos. Energy efficient building ventilation control using the finite element modelling of SRC. *IEEE European Modelling Symposium*, Malta 14-16 Nov. 2012.

## Appendix A: Querying of IFC models and sample query

For the development of IFC queries a stand-alone tool is used basing on Java code and performed in an Eclipse environment (EMS Eclipse Modelling Framework). The implementation of libraries included in the BIMserver package enables the access to the BIMserver, loads (deserialises) an IFC file into an object variable and can be used to define different logic for certain data request. The use of the BIM libraries and the data model allows a comfortable handling of the data but is limited to the state of completeness of the implantation regarding server and the IFC version.

As described in the introduction of this section the tools used for the development of queries depend on the actual state of versions. The following description is based on IFC2x3 and BIMServer 1.1.

### A.1 IFC data access - sample query

This section describes the data access and querying method using a simple example. The problems found within the developing process are also mentioned and described, mostly due to problems with versions (either IFC or BIMServer).

The main target of the sample query is getting access to the content of the IFC model, i.e. components, properties, relations etc. The tool used in this sample is a local BIMServer (actual 1.1, the next step will be done with a modified BIMServer 1.2 as described in D2.3). The libraries of the BIMServer base currently on IFC2x3, with the new server they will be switched to IFC4.

The tool for developing the query code is based on Java within the IDE Eclipse. The sample query will focus on the data related to the HVAC system. For the building model the export interface of Revit is used to generate the IFC file.

The main problem that occurs together with BIMServer 1.1 is the absence of values for Property Sets since they contain most of the required data. As will be seen in the sample below they are not accessible with BIMServer 1.1. The switch over to BIMServer 1.2 hopefully will solve this problem without producing others.

### A.2 Task for the sample query

The underlying building model for the sample query is a standalone heating system without connections to other systems, e.g. no building geometry data, no connection to weather file, no link to sensors and control signals (APOs). Thus the querying task focusses on static data from the IFC file, not on operating data.

The data source is also not linked to the complex BaaS system since within the developing process the system is not yet established and the separation of data systems avoids additional problems with data exchanges within the design process of queries. The system will be merged in the terminating phase of the project. The BIMserver for the sample query is running standalone and locally.

The task of the sample query can be summarised in following steps

- 1) Establish connection to BIMServer and load model
- 2) Get structure of IFC model by analysing relations
  - a) Parse model and find generator (starting point)

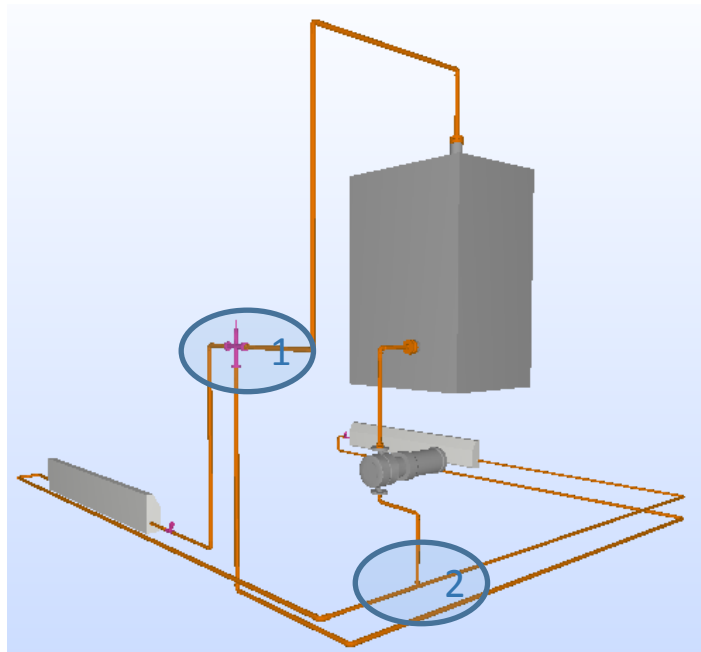
- b) Find chain of connected distribution objects (in direction of flow) and follow the path until end is reached, i.e. return to generator
  - c) On the way: analyse the model structure and collect relevant data (properties) required for the simulation model, e.g. pipe length, diameter, boiler power, ...
- 3) Transfer to a corresponding simulation model
    - a) Find simulation components corresponding to building components
    - b) Add auxiliary components and simulation boundary conditions
    - c) Fill missing data with default values (previously defined)
    - d) Generate input file for simulation engine
  - 4) Run simulation and get results

### A.3 Sample IFC model

The sample to be queried is displayed in the next figure and contains one boiler, one pump and two radiators. The query has to manage (in direction of flow)

- the 3 Way Valve (marked with "1") and divide the path into
- section 1: terminating at radiator 1 and
- section 2: terminating at radiator 2
- then collected in the tee-piece (marked with "2")
- follow up to the pump
- and terminate at boiler inlet

The next figure displays the boiler circuit with the distribution and emission components.



**Figure 39: Boiler circuit with 3-way valve and tee piece**



#### A.4 Java - Establish connection to BIMServer and load model

The following table contains extractions (separated by "...") of the most important sections of the Java code for this task. For the connection and the access to the BIMserver classes are used provided by the BIMServer library. The green text with prefix "//" comments and explains the different code steps.

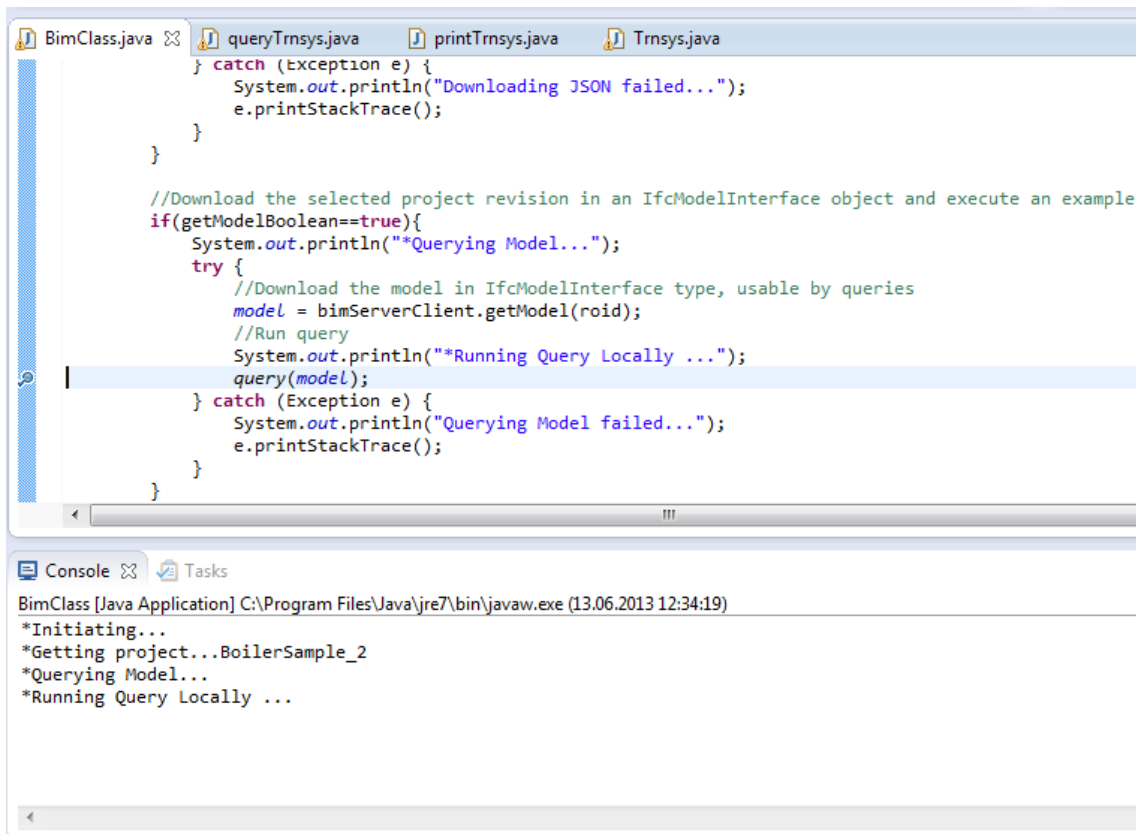
Compared to the use of a browser based interface of the BIMServer, the first section is the equivalent of the login page using <bimServerClient>. The second section contains the statements regarding the access to a certain project/revision and gets the ID of the project <roid>. In the most important third section the IFC file of the project is loaded (deserialised) into <model>, so this variable now contains the structure of the whole IFC-file and can be hand over to the querying routine.

##### Listing 1: Java code extractions - BIMServer connection

```
...
//Establish Connection via Protocol Buffers. Change the factory to connect
with soap
org.bimserver.plugins.PluginManager pluginManager = new
org.bimserver.plugins.PluginManager();
pluginManager.loadPluginsFromCurrentClassLoader();
//Set the IP of the server (e.g. localhost) and the protocol buffers port
(usually 8020)
ProtocolBuffersBimServerClientFactory factory = new
ProtocolBuffersBimServerClientFactory(hostIP, 8020, pluginManager);
UsernamePasswordAuthenticationInfo authenticationInfo = new
UsernamePasswordAuthenticationInfo(username, userpw);
//Login to Server and establish Connection
BimServerClient bimServerClient = factory.create(authenticationInfo,
hostIP);
...
System.out.println("Getting project..." + projectName);
long roid = -1;
try {
    //Get the project
    SProject_list=bimServerClient.getServiceInterface().getProjectsByName(pro
jectName);
    //Get the first element of the list (our project)
    SProject_test= SProject_list.get(0);
    //Get last revision of the project
    roid =SProject_test.getLastRevisionId();
} catch (Exception e) {
    System.out.println("Getting project failed...");
    e.printStackTrace();
}
...
//Download the selected project revision in an IfcModelInterface object and
execute an example query
if(getModelBoolean==true){
    System.out.println("Querying Model...");
    try {
```

```
//Download the model in IfcModelInterface type, usable by queries
model = bimServerClient.getModel(roid);
//Run query
System.out.println("Running Query Locally ...");
query(model);
} catch (Exception e) {
    System.out.println("Querying Model failed...");
    e.printStackTrace();
}
}
...
//Example query
public static void query(IfcModelInterface model) {
    System.out.println("Running example");
    initializeModelInverseSlots(model);
    query_objects(model);
}
```

The following figure is a screenshot of the Eclipse environment where the console at the bottom contains the print instructions of the code steps during the execution.



**Figure 40: Screenshot of Java tool during execution**

## A.5 Java - Get structure of IFC model by analysing relations

In order to detect the structure of the model there are two steps to find the chain using the information about the relations among the components, i.e.

- a) Parse model and find generator (starting point).
- b) Find chain of connected distribution objects (in direction of flow) and follow the path until end is reached, i.e. return to generator.

The sub routine includes the access to a collection of objects (<getAllWithSubTypes> generates a List with all the objects in this model of the given type, instances of subtypes are included), in order to find the boiler, where temporarily a workaround is used.

The sub routine <Chain> is recursive, i.e. the function calls itself to build up a chain of "next elements" with increasing recursion depth. The statements at the beginning handle the reaction on special events, i.e. in case of section spitting if number of ports greater then 1 or in case the end of a section is reached.

### Listing 2: Java code extractions - access to model structure

```
...
private static void query_objects(IfcModelInterface model){
    System.out.println("query objects ...");
    // get all objects => boiler, pump, radiators
    List<IfcBuildingElementProxy> objects =
        model.getAllWithSubTypes(IfcBuildingElementProxy.class);
    // loop objects
    for (IfcBuildingElementProxy object : objects){
        // workaround: get boiler by name (we know that "Boiler" is part of the
        name)
        if ((object.getName().contains("Boiler"))){
            System.out.println("Start = " + object.getName());
            // sub: get chain
            Chain(object, 0);
        }
    }
}
...
private static void Chain(IfcElement element, int k){
    // helper function => information about the existance of ports for the
    element
    if (list_NextElements(element).size() > 1){
        k++;
        System.out.println("    #section " + k + ":");
    }
    // is end of section reached? (using workaround with part of radiator name,
    like boiler workaround)
    if (element.getName().contains("Hydronic Fin Tube")){
        k++;
        System.out.println("    #section " + k + ":");
        return;
    }
}
```

```
List<IfcElement> nextelements = list_NextElements(element);
for (IfcElement nextelement : nextelements) {
    String out = "    next: " + nextelement.getName();
    System.out.println(out);
    Chain(nextelement, k);
}
}
...
// sub function as helper => information about the existence of ports for the
// element
private static List<IfcElement> list_NextElements(IfcElement element){
    List<IfcElement> foundElements = new ArrayList<IfcElement>();
    if (element.getHasPorts().size() > 0){
        for (IfcRelConnectsPortToElement connects : element.getHasPorts()) {
            IfcPort port = connects.getRelatingPort();
            if (port.getConnectedTo().size() > 0) {
                for (IfcRelConnectsPorts portconnects1 : port.getConnectedTo()){
                    for (IfcRelConnectsPorts portconnects2 :
portconnects1.getRelatingPort().getConnectedTo()){

                        foundElements.add(portconnects2.getRelatedPort().getContainedIn().getRelatedElement());
                    }
                }
            }
        }
    }
    if (foundElements.size() < 1){
        foundElements.add(element);
    }
    return foundElements;
}
```

The result of the simple query is just the printout of object names to the console, only to show that the recursive approach is able to detect the structure of a chain of elements and also manages the section splitting.

#### Console:

```
Initiating...
Getting project...BoilerSample_2
Querying Model...
Running Query Locally ...
Running example
query objects ...
Start = Boiler:Standard:Standard:651755
    next: Reducer Straight - Soldered - CU:Standard:Standard:688707
    next: Pipe Types:Standard:688695
    next: Elbow - Soldered - CU:Standard:Standard:688962
    next: Pipe Types:Standard:688635
```

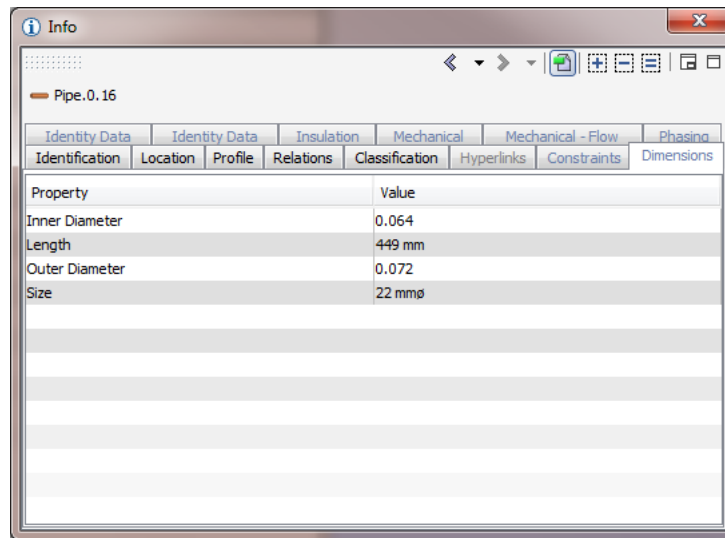
```
next: Elbow - Soldered - CU:Standard:Standard:688966
next: Pipe Types:Standard:688520
next: Elbow - Soldered - CU:Standard:Standard:688522
next: Pipe Types:Standard:653187
next: Reducer Straight - Soldered - CU:Standard:Standard:653209
next: 3 Way Valve - 20-100 mm:25 mm:25 mm:652848
#section 1:
next: Reducer Straight - Soldered - CU:Standard:Standard:659688
next: Pipe Types:Standard:653091
next: Elbow - Soldered - CU:Standard:Standard:659701
next: Pipe Types:Standard:659693
next: Reducer Straight - Soldered - CU:Standard:Standard:688146
next: Elbow - Soldered - CU:Standard:Standard:688144
next: Pipe Types:Standard:690051
next: Reducer Straight - Soldered - CU:Standard:Standard:690065
next: Radiator Valve - 15-22 mm - Threaded:15 mm:15 mm:689345
next: Reducer Straight - Soldered - CU:Standard:Standard:690042
next: Pipe Types:Standard:690028
next: Reducer Straight - Soldered - CU:Standard:Standard:690044
next: Radiator - Hydronic Fin Tube:20 NPT:20 NPT:651938
#section 2:
next: Pipe Types:Standard:686678
next: Reducer Straight - Soldered - CU:Standard:Standard:686706
next: Pipe Types:Standard:686604
next: Elbow - Soldered - CU:Standard:Standard:686708
next: Pipe Types:Standard:653320
next: Elbow - Soldered - CU:Standard:Standard:653473
next: Pipe Types:Standard:653460
next: Elbow - Soldered - CU:Standard:Standard:653497
next: Pipe Types:Standard:653477
next: Elbow - Soldered - CU:Standard:Standard:659583
next: Pipe Types:Standard:659575
next: Reducer Straight - Soldered - CU:Standard:Standard:689977
next: Elbow - Soldered - CU:Standard:Standard:689975
next: Pipe Types:Standard:689953
next: Reducer Straight - Soldered - CU:Standard:Standard:689973
next: Radiator Valve - 15-22 mm - Threaded:15 mm:15 mm:689780
next: Reducer Straight - Soldered - CU:Standard:Standard:689997
next: Pipe Types:Standard:689983
next: Reducer Straight - Soldered - CU:Standard:Standard:686328
next: Radiator - Hydronic Fin Tube:20 NPT:20 NPT:651985
# Done in 12 s
```

The next task of the routine comprises the access to the object information where the previous task was focussed on object relations, i.e.

- c) On the way: analyse the model structure and collect relevant data (properties) required for the simulation model, e.g. pipe length, diameter, boiler power, ...

Most of the IFC Property Sets contain generic data similar among most of the different objects, i.e. Identification (Name), Location and Dimensions. Additional Property Sets are related to type specific data. As an example the content of a pipe (Entity <IfcPipe>) is displayed in the

following figure using a screenshot of an IFC viewer. The properties within the Property Set "Dimensions" are: inner diameter, length, outer diameter and size.



**Figure 41: Screenshot of IFC-file in 'Solibri': values of PropertySet "Dimensions"**

The output of a sample query regarding properties on the console looks like (section for the pipe)

```
...
next: Pipe Types:Standard:688695
PropertySet 'Dimensions':
    Size = null / Inner Diameter = null / Length = null / Outer Diameter = null /
```

The IFC file contains the information since the IFC viewer could not display the values if they were not defined but the query returns 'null' for all values.

It seems that the deserialization of the BIM library does not include the PropertySets. It is expected that in the new version of IFC and BIMServer this problem will be solved.

Extractions of the corresponding Java code are contained in the following listing. The code has to get access to the property sets (via relations), to check whether they are defined and returns the property names and values contained in the property set.

**Listing 3: Java code extractions - properties**

```
...
str_GetProperties(nextelement, "Dimensions");
...
// sub function as helper => get property set by name
private static String str_GetProperties(IfcElement element, String
PropertySetName) {
    String returnstring = "PropertySet '" + PropertySetName + "': ";
    for (IfcRelDefines isDefined : element.getIsDefinedBy()){
        if (isDefined instanceof IfcRelDefinesByProperties){
            IfcRelDefinesByProperties isDefinedProperty =
                (IfcRelDefinesByProperties)isDefined;
```

```
    IfcPropertySetDefinition relProperty =
isDefinedProperty.getRelatingPropertyDefinition();
    if (relProperty.getName().equals(PropertySetName)){
        IfcPropertySet Properties = (IfcPropertySet)relProperty;
        for (IfcProperty Property : Properties.getHasProperties()){
            IfcPropertySingleValue PropertyValue =
(IFcPropertySingleValue)Property;
            IfcValue Value = PropertyValue.getNominalValue();
                returnstring += Property.getName() + " = " + Value + " / ";
        }
    }
}
return returnstring;
}
```

#### A.6 Java - Transfer to a corresponding simulation model

One of the outcomes of the development of simulation modelling tools is the identification and definition of relations between simulation components including their parameters and components - or more abstract: elements - from different data sources. In case of static data from the IFC file (basic knowledge about the components), a catalogue will be developed containing corresponding components in order to transfer the IFC model into a simulation model, e.g. boilers, pumps, radiators and distribution elements. The process that includes the identification of the type of IFC objects to be looked up in the catalogue in order to get the corresponding simulation component (in case of this example TRNSYS component). The parameter and relations of the simulation will be filled with the information found in the corresponding IFC file or from other data sources, such as weather data. Additional components for the simulation contain general simulation boundaries are also required even if they have no correspondence in the IFC model. The task includes:

- a) Find simulation components corresponding to building components
- b) Add extra components and simulation boundary conditions
- c) Fill missing data with default values (previously defined)

See section 5.5 for further information.

For the testing of the queries data access, data exchange and data processing a temporary solution was developed as a workaround since the BaaS system - where the individual operations later on will be automatically and not visible - is not yet established. The temporary workflow within the developing process includes the generation of a (text based) input file for the simulation engine and as result the receipt of a (text based) output file of the simulation engine. The first step contains to

- d) Generate input file for simulation engine

The output is displayed in the next figure where the screenshot shows the console Eclipse containing the content for the simulation input file. The code for the input file generation bases on the constraints given by the simulation engine, i.e. structural requirements of the text file.

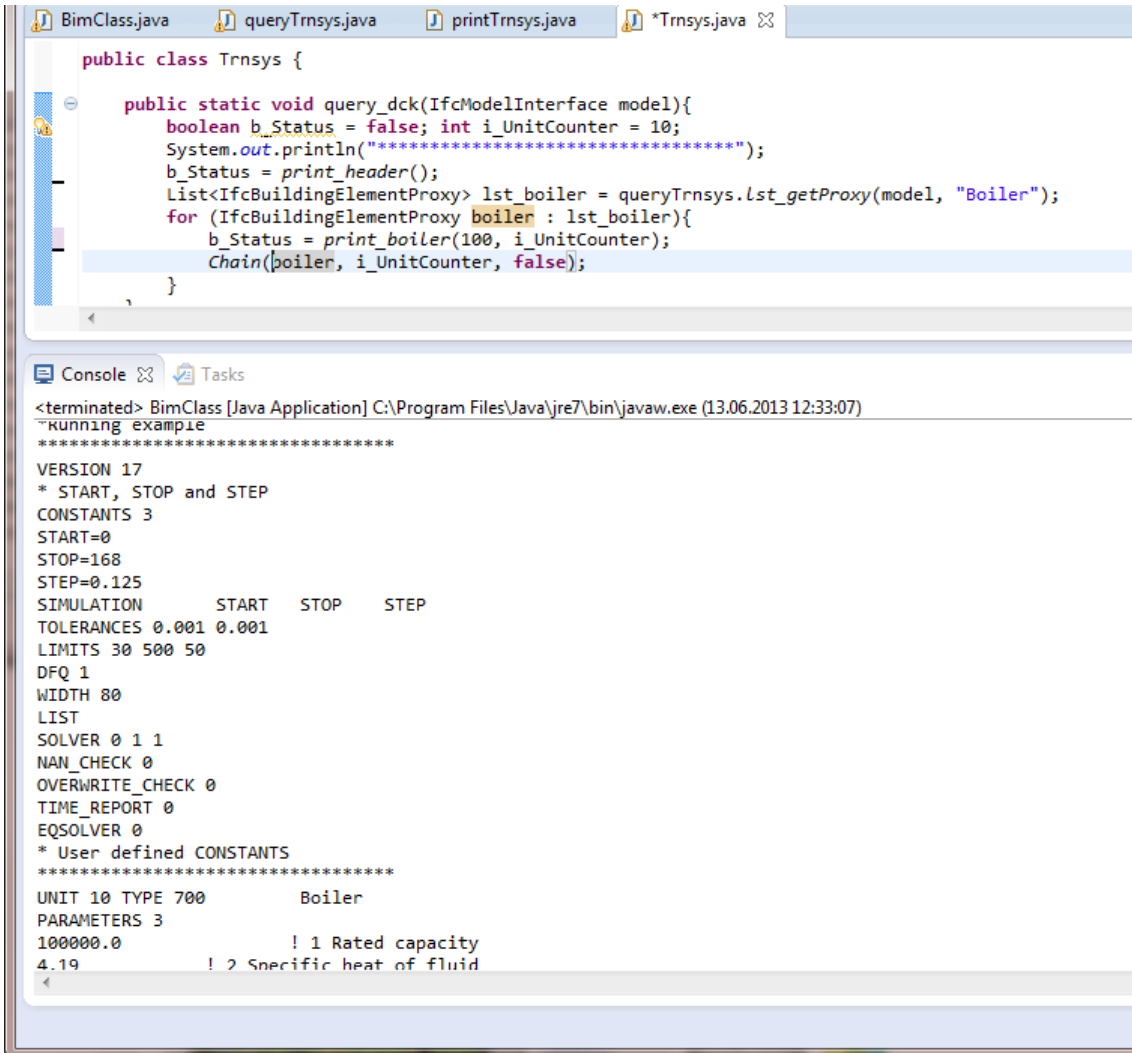
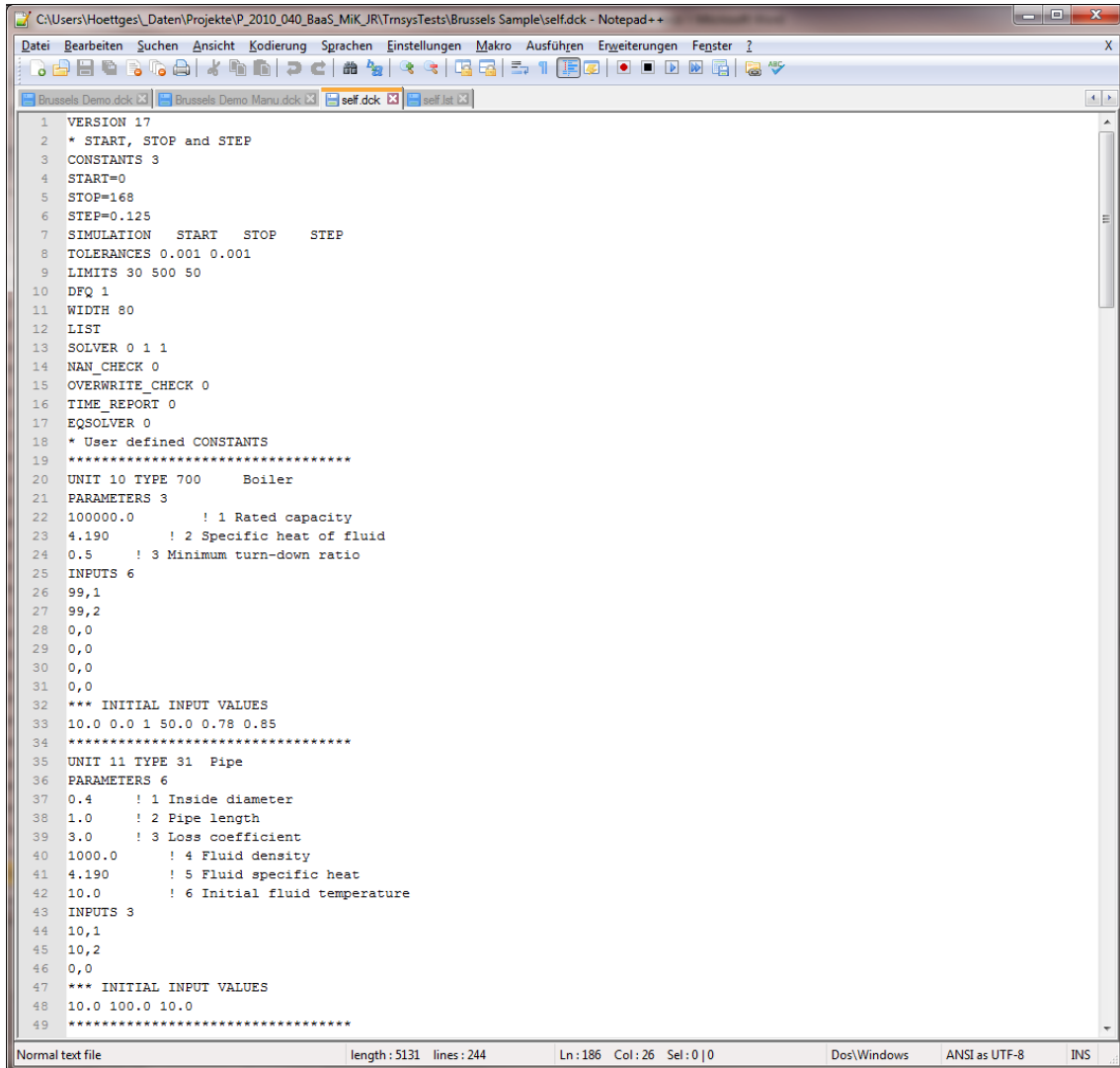


Figure 42: Screenshot of Java tool during execution - generation of simulation input file



The next step is a simple workaround, i.e. to manual copy the text from the output console to a txt-file. The following screenshot displays the file opened in an editor tool.



```

1 VERSION 17
2 * START, STOP and STEP
3 CONSTANTS 3
4 START=0
5 STOP=168
6 STEP=0.125
7 SIMULATION START STOP STEP
8 TOLERANCES 0.001 0.001
9 LIMITS 30 500 50
10 DFQ 1
11 WIDTH 80
12 LIST
13 SOLVER 0 1 1
14 NAN_CHECK 0
15 OVERWRITE_CHECK 0
16 TIME_REPORT 0
17 EQSOLVER 0
18 * User defined CONSTANTS
19 *****
20 UNIT 10 TYPE 700 Boiler
21 PARAMETERS 3
22 100000.0 ! 1 Rated capacity
23 4.190 ! 2 Specific heat of fluid
24 0.5 ! 3 Minimum turn-down ratio
25 INPUTS 6
26 99,1
27 99,2
28 0,0
29 0,0
30 0,0
31 0,0
32 *** INITIAL INPUT VALUES
33 10.0 0.0 1 50.0 0.78 0.85
34 *****
35 UNIT 11 TYPE 31 Pipe
36 PARAMETERS 6
37 0.4 ! 1 Inside diameter
38 1.0 ! 2 Pipe length
39 3.0 ! 3 Loss coefficient
40 1000.0 ! 4 Fluid density
41 4.190 ! 5 Fluid specific heat
42 10.0 ! 6 Initial fluid temperature
43 INPUTS 3
44 10,1
45 10,2
46 0,0
47 *** INITIAL INPUT VALUES
48 10.0 100.0 10.0
49 *****
  
```

**Figure 43: Screenshot of editor - simulation input file**

### A.7 Java - Run simulation and get results

In case of the TRNSYS simulation engine the auxiliary tool "TRNEdit" allows the standalone execution of a simulation run without using the GUI of TRNSYS (Simulation Studio). The next two figures contain a screenshot of TRNEdit after opening of the input file and just before start of simulation "Calculate". The result is contained in the second figure where the simulation output chart represents the successful execution of the simulation run.

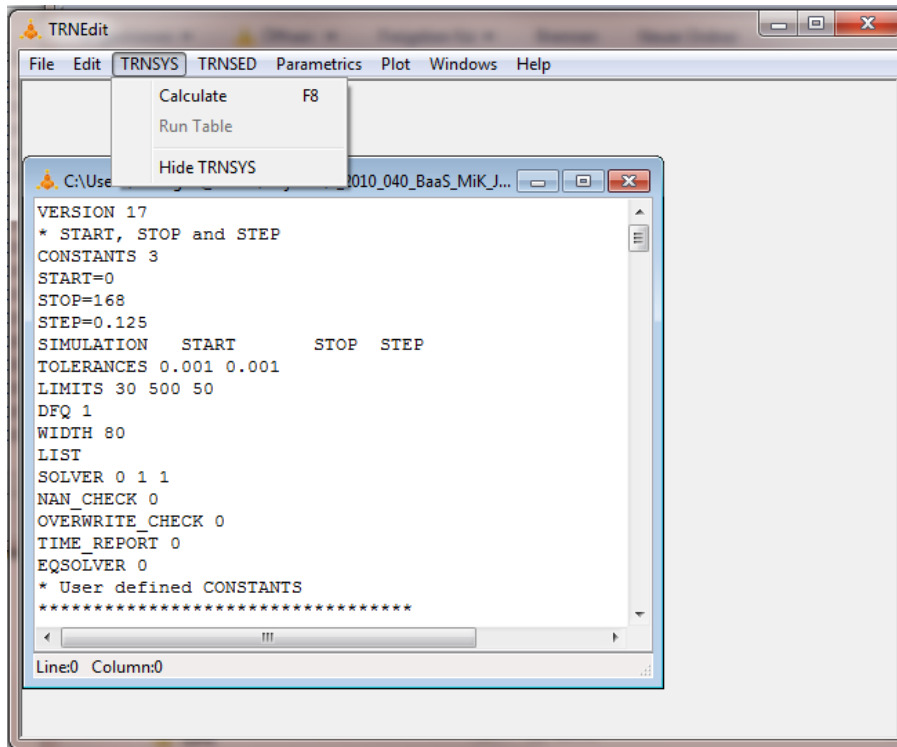


Figure 44: Screenshot of simulation engine

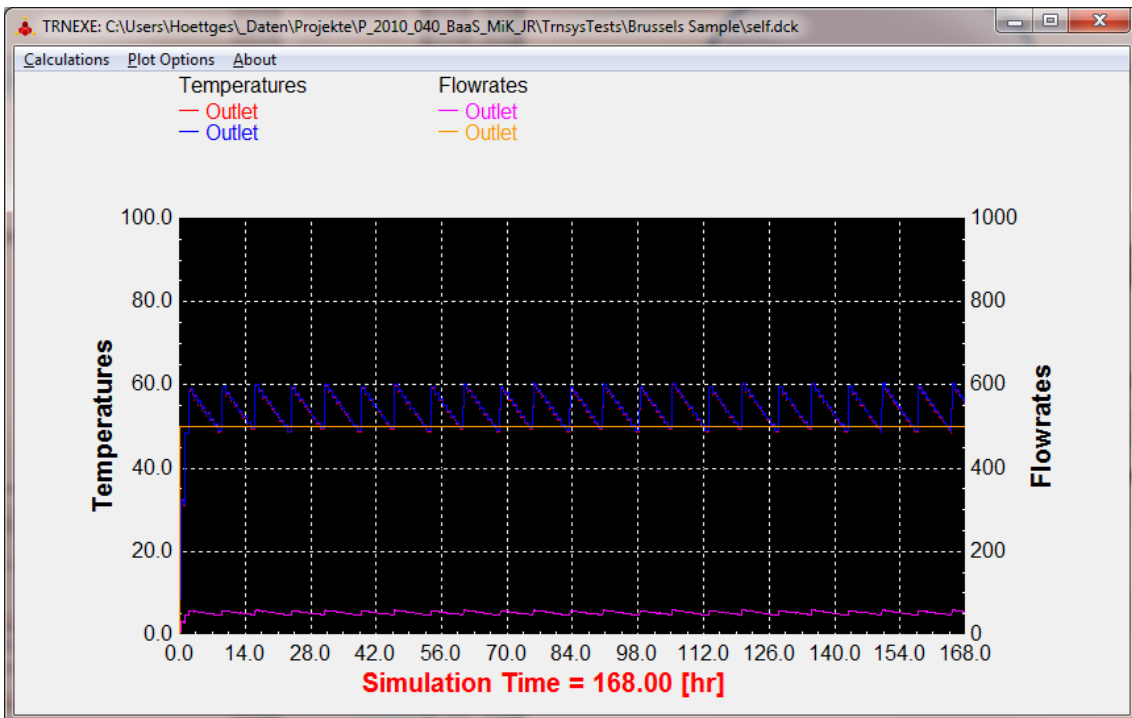
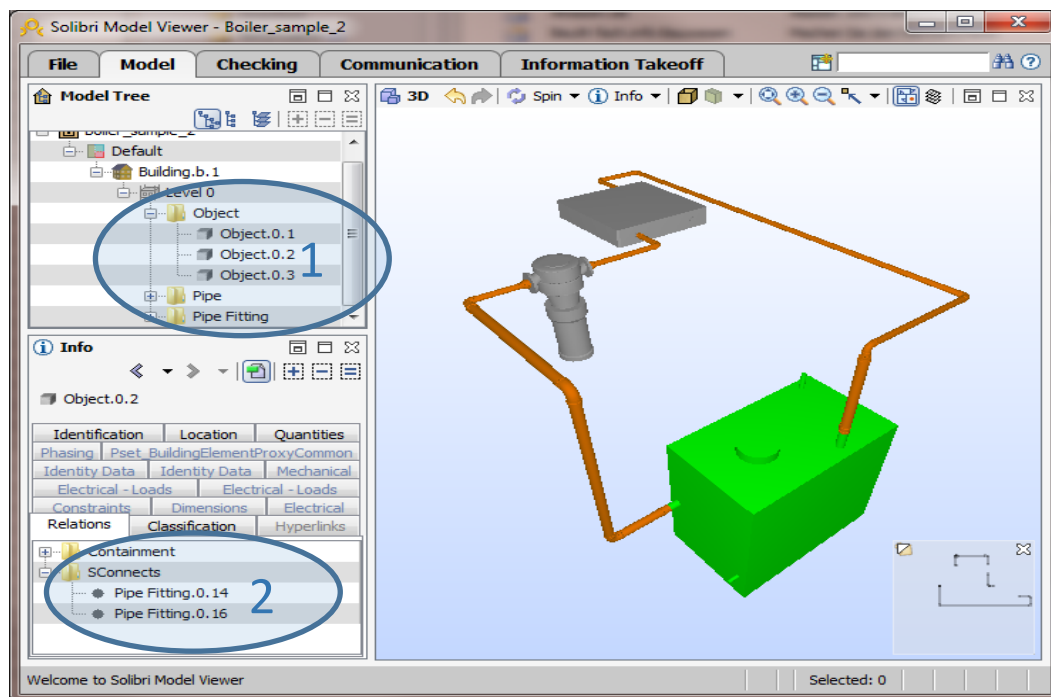


Figure 45: Screenshot of simulation result

## A.8 Problems occur with the sample query

### Find chain from Boiler to Radiators

- pipes and pipe fittings are (proper) modelled as IFCDistributionElements
- and connected via IFCDistributionPorts, distinguishing “sink” and “source” in order to define the direction of flow
- but: Radiator, Boiler and Pump are modelled as “objects” using class IfcBuildingElementProxy as a placeholder see figure marked with „1“
- and Revit does not use the port object for establishing the relations, i.e. they have to be added manually in the ifc-file see figure marked with „2“



**Figure 46: Boiler circuit and relations**

Sample of manual manipulation of ifc-file (sink and source port of one distribution element), change in editor:

```
#80010= IFCRELCONNECTSPORTTOELEMENT('ConnectsPortToElement#80010',#52,$,$,#80011,#37695);
#80011= IFCDISTRIBUTIONPORT('DistributionPort#80011',#52,$,$,'IfcDistributionPort',$,$,.SINK.);
#80012= IFCRELCONNECTSPORTS('RelConnectsPorts#80012',#52,$,$,#80011,#80013,$);
#80013= IFCDISTRIBUTIONPORT('DistributionPort#80013',#52,$,$,'IfcDistributionPort',$,$,.SOURCE.);
#80014= IFCRELCONNECTSPORTTOELEMENT('ConnectsPortToElement#80014',#52,$,$,#80013,#479);
```

```
#80010= IFCRELCONNECTSPORTTOELEMENT('GlobalUniqueNameOfComponentA',#52,$,$,#80011,#37695);
#80011= IFCDISTRIBUTIONPORT('DistributionPort#80011',#52,$,$,'IfcDistributionPort',$,$,.SINK.);
#80012= IFCRELCONNECTSPORTS('RelConnectsPorts#80012',#52,$,$,#80011,#80013,$);
#80013= IFCDISTRIBUTIONPORT('DistributionPort#80013',#52,$,$,'IfcDistributionPort',$,$,.SOURCE.);
```

---

```
#80014= IFCRELCONNECTSPORTTOELEMENT ('GlobalUniqueNameOfComponentB', #52, $, $, #80013, #479);
```

Since the boiler is not modelled as a "proper" boiler entity, we need a workaround to detect the boiler object, i.e. get boiler by name (we know that "Boiler" is part of the name).

**Problem: Property Sets are there but values are 'null'**

It seems that the deserialization of the BIM library does not include the PropertySets. If this is true and will not be solved in the next updates, we must switch the query tool (to direct access via STEP).