D5.5

Parametric modelling techniques for EeB

Deliverable Report D5.5 Final version

STREAMER - Optimised design methodologies for energy-efficient buildings integrated in the neighbourhood energy systems.
The STREAMER project is co-financed by the European Commission under the seventh research framework programme FP7.EeB.NMP.2013-5 ; GA No. 608739)

D5.5

Parametric modelling techniques for EeB

Main author Stefan van Nederpelt

Co-authors Matthias Weise, Karl-Heinz Haefele, Steffen Hempel, Joachim Benner, Guillaume Picinbono, Jan-Peter Pols.

Dissemination PU

Document history

Version	Date	Status	Produced by	Comments
0.8	2015.08.18	Draft	S. van Nederpelt	Reviewed by WP leader
0.9	2015-08-21	Draft	S. van Nederpelt	Reviewed draft
1.0	2015-08-28	Final draft	S. van Nederpelt	Reviewed by Technical Committee
		Reviewed draft		
1.2	2015-09-03	Approved	Freek Bomhof	
		Submitted		

Colophon

The STREAMER project is co-financed by the European Commission under the seventh research framework programme with contract No.: 608739 - FP7-2013-NMP-ENV-EeB. The information in this publication does not necessarily represent the view of the European Commission. The European Commission shall not in any way be liable or responsible for the use of any such knowledge, information or data, or of the consequences thereof.

Abstract

In the STREAMER description of work, this deliverable is described as:
"Parametric modelling techniques for EeB: Semantic Web PMO format that shows the parametric extension of IFC towards storage of the parametric knowledge in an open format. There is a close relation with the configurator of parametric solutions developed in WP6."

A general issue for STREAMER is that no existing BIM tool supports a standard for importing and exporting parametric BIM models. The IFC standard, being supported by all major BIM tools and used as central format for exchanging BIM data in STREAMER, is not able to represent advanced parametric models. The development of "Parametric IFC" has started (see chapter 2), but is still in a state where it is neither supported by any existing BIM tool nor can be used for one of the design tools prototypically developed in STREAMER WP6. Thus, the usage of traditional parametric modelling techniques and tools in STREAMER is limited.

The intended goal of this deliverable is to support the EDC to translate a Programme of Requirements (PoR), into a spatial arrangement. This is the main function of the EDC. The EDC demands a set of general design rules which represent the "tacit knowledge" of design experts to be able to translate a PoR into design proposals. These design rules are imported into the EDC.

For reasons mentioned above, the scope of this deliverable has been extended with the development of design rules. The deliverable will thus focus on the current state-of-art in parametric modelling, the possibilities of using parametric modelling in STREAMER, AND design rules.

Publishable executive summary

Central topic of this deliverable is the usage of parametric modelling techniques for designing energy efficient buildings. The main feature of a parametric building model is that the geometry is defined by a set of parameters and corresponding mathematical functions, enabling the generation of the explicit geometry according to the used parameter values.

A general issue for STREAMER is that no existing BIM tool supports a standard for importing and exporting parametric BIM models. The IFC standard, being supported by all major BIM tools and used as central format for exchanging BIM data in STREAMER, is not able to represent advanced parametric models. The development of “Parametric IFC” has started, but is still in a state where it is neither supported by any existing BIM tool nor can be used for one of the design tools prototypically developed in STREAMER WP 6. Thus, the usage of traditional parametric modelling techniques and tools in STREAMER is limited.

The intended goal of this deliverable is to support the EDC to translate a Programme of Requirements (PoR), into a spatial arrangement of rooms and functional areas within a given building shape. This is the main function of the EDC. The EDC demands a set of general design rules which represent the “tacit knowledge” of design experts to be able to translate a PoR into design proposals. These design rules are imported into the EDC.

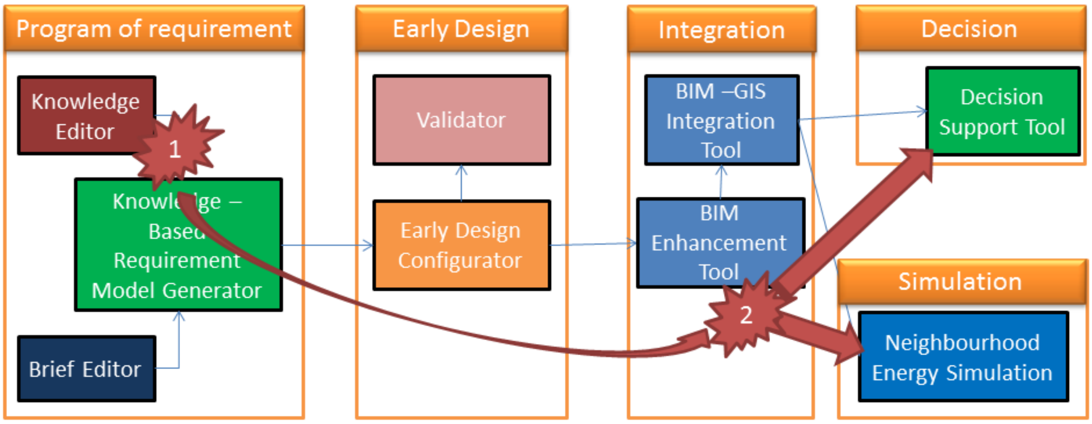
For reasons mentioned above, the scope of this deliverable has been extended with the development of design rules. The deliverable will thus focus on the current state-of-art in parametric modelling, the possibilities of using parametric modelling in STREAMER, AND design rules.

In the AEC sector, parametric modelling should not be limited to geometric modelling. It can be used for almost all information that constitutes a complete building model:

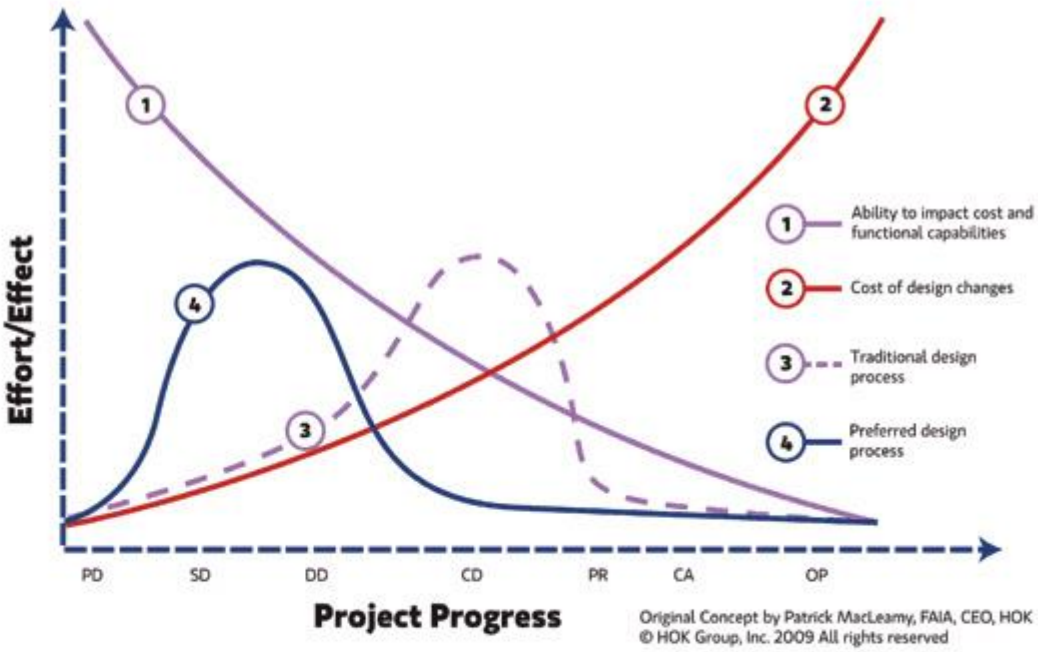
- Form (shape & size)
- Attributes (materials & physical properties)
- Relationship/Assembly (how parts are related, what moves with what)
- Behavior (doors opening and closing, structural members under load)

Propositions for the level of parametric modelling to be included in STREAMER:

- Keep a pragmatic approach, by choosing to use "generator of IFC" when sufficient: for this purpose, tools can deal with input information (modelling it, checking it, using it), but will not share it with other tools.
- Identify some strategic Phase/Scale in which "Key parametric information" would need to be shared using IFC. In these cases, experiment implementation of IFC parametric modelling.
- Consider more advanced use of Parametric IFC for products and component systems (parametric IFC catalog model).



The STREAMER project possesses a very interesting element that adds another dimension to this deliverable: design automation. The design rules are intended to be used by the EDC. The EDC creates an optimized spatial configuration of required spaces by using design rules. The design rules can be prioritized by providing KPI settings within the EDC, allowing the creation of multiple designs, all using the same design rules.



In the STREAMER DOW, the previous image above plays an important role. In the envisioned process, the design effort (line #4) peaks in the early design, where it has a high impact on the design performance (line #1) at the lowest cost (line#2). The availability of design rules and the expert knowledge that it represents is expected to help designers to make design decisions from a well-informed position and within a limited amount of time in this early design phase.

Design rules as used by architects are based on many factors which include expert knowledge, personal preferences, personal experiences, conventional wisdom, regulations, cultural context and medical requirements. Although design rules contained in building regulation can easily be found, the documentation of design rules by designers seldom occurs. Due to the dependency on STREAMER-specific attributes to which the design rules are applied, the design rules in this deliverable have been created from scratch.

Due to the context-specific nature of the factors underlying the design rules, it is neither achievable nor desirable to produce an optimum or complete set of design rules that can be used in every situation. Therefore the primary focus has been on general design rules which are not too project-, person- or location-specific and with relevance in the early design phase.

Although the design rules have been created as input for the EDC, their presentation in natural language allows them to also be used in a manual (human) process, for example to validate an existing design. The design team can use this general, predefined set of design rules in a format that allows changes or additions to be easily made for a specific situation. In the design rules developed in this deliverable, non-negotiable rules have been clearly identified, although they can be modified. For example, the maximum evacuation length is part of building regulation, although the maximum length varies between countries.

Because the EDC does not use favoritism like humans do, some interesting / surprising results might be expected, although at the time of writing it’s too early to tell whether the quality of these results is good enough to be seriously considered in the design process.

However, developments in other fields indicate that application of tools like the EDC can be of added value to the design. Ship designers in The Royal Netherlands Navy, for example, use a design methodology called the “packing approach”. Like the EDC, this approach is also based on design rules and a PoR.

The following image shows several examples of early ship configurations as created by the “packing approach” (2011, Van Oers):

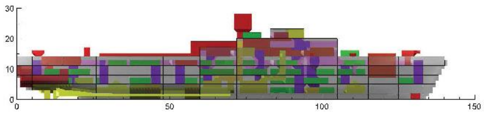


Fig. 6.40 – Multi-mission frigate: design no. 12964

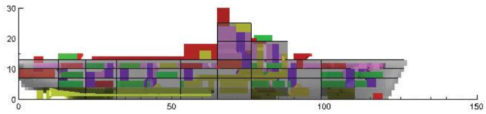


Fig. 6.44 – ASW-frigate: design no. 10686

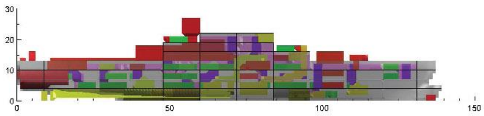


Fig. 6.41 – Multi-mission frigate: design no. 16748

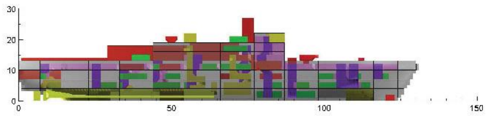


Fig. 6.45 – ASW-frigate: design no. 2591

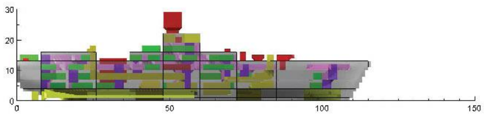


Fig. 6.42 – AAW-frigate: design no. 4200

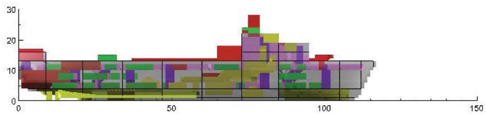


Fig. 6.46 – OPV: design no. 26298

Fig. 6.43 – AAW-frigate: design no. 9120

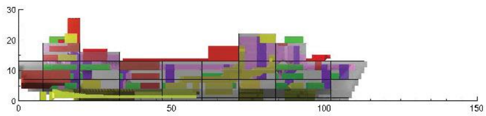


Fig. 6.47 – OPV: design no. 28923

As can be seen by looking at the design numbers mentioned below the results in this figure, computers are able to create huge amounts of design alternatives. This immediately creates a problem, because who is going to review all these alternatives and select the most suitable designs? Clearly, when designers generate this many design alternatives, they need to be supported by validation tools. In STREAMER deliverable 3.6, a decision-support tool will be developed in which the ‘performance’ of these design alternatives is displayed on a dashboard, based on the STREAMER KPIs.

The design rules either relate two different spatial objects (one to one), or specify a form of clustering (one to many). The relation between the KPIs and the design rules is very important for the EDC. In the EDC, the design team is expected to control which KPIs should be fulfilled with the highest priority, which will lead to a specific design output by the EDC. Even with a limited amount of design rules, many different design outputs can be generated with different KPI priorities. And when design rules are conflicting, the KPIs can determine which design rule has the highest priority.

Due to the possible occurrence of contradicting design rules, it is unlikely that all rules can be fully observed simultaneously. This situation will especially occur when rules concentrate on single, conflicting Key Performance Indicators of the design (e.g. energy efficiency, cost efficiency or operational efficiency, see Deliverable D 3.1 “Building-oriented EeB KPIs of newly designed and retrofitted buildings”). By varying the priority of these design rules, the EDC in its final version will be able to generate early design alternatives with a user-defined weighting of KPIs.

In D1.3, the conclusion was drawn that most building envelope solutions have no relation with the activities inside the hospital (represented by the labels). So far, only a couple of relations have been identified, such as: windows vs. daylight requirement, or natural ventilation system vs. the presence of vents.

At the moment of writing, the EDC is not intended to generate building envelope solutions. However, these relations have been anticipated in the design rules. Assuming that it is possible to make a window or a vent in the facade manually in a later design phase, functional areas, rooms and spaces requiring daylight or natural ventilation are simply placed at a distance of 0 to the edge of the building mass, where the facade will be. This is a general design rule (not project specific).

An envisioned additional functionality of the EDC is the addition of HVAC component system suggestions to functional areas and rooms. These suggestions are made by using a method similar to the one first introduced in D1.3. In short: by comparing label values of functional areas or rooms to label values of HVAC component systems, a selection of compatible HVAC systems can be made for each functional area or room. At this moment, the EDC does not support this feature, although it is expected that it can be incorporated based on the method described in this paragraph. At this moment, only the most energy efficient HVAC systems are prioritized. When finance and quality KPI performance of the HVAC systems are also known, it will be possible to integrate the systems choice with these KPIs as well. Depending on the progress with the EDC and integration with WP2, this can be researched in D5.6.

The Knowledge Editor (developed in D6.1) provides a specific formal language (so called Domain Specific Language" of DSL) to represent rules, an a specialized editor to generate DSL-rules. This language is similar to your rule formulation in structured natural language, but not equal. The user of the Knowledge Editor informally uses rule formulations to generate new representations in DSL. Only the transformation DSL → XML is done automatically. More information about the knowledge editor is provided in deliverable D6.1.

List of acronyms and abbreviations

BIM : Building Information Modelling

CSV: Comma Separated Value

DoW: Description of Work, Annex 1 to the STREAMER grant agreement

EeB : Energy efficient Buildings

EDC : Early Design Configurator

PMO: Product Modelling Ontology

HVAC : Heating, Ventilation, Air Conditioning

IFC : Industry Foundation Classes

KPI: Key Performance Indicator

MEP : Mechanical, Electrical, Plumbing technologies

PoR : Programme of Requirements

XML: eXtensible Markup Language

Definitions

To create a common language and prevent misunderstanding it is important to have common terms and definitions (glossary):

Parameter: A parameter is an input for an algorithm. The output of the algorithm is changed by the input of the parameter. When using the standard math term f(x) then f is the name of the algorithm, x is name of the parameter. Examples of parameters are: dimensions used to create model geometrical representation, material, material properties, formulas to compute some values used in the model.

Attribute: An attribute is something where you can say “X has Y” where Y is the attribute. e.g. “has color”, “has wheel”. X is an object. When X is actually existing, the attribute has a value. e.g. “The grass on my front porch has color green”,

Parametric modelling: Parametric modelling uses parameters to define a model.

Parametric extension of IFC format: This extension provides material to express the fact that attributes of IFC objects are computed from parameters and algorithms. An example is PA-1 “Parametric IFC”, a proposal of IFC model extension aiming at “allowing any aspect of an IFC model to be driven parametrically”. It was proposed by AEC3 and TUM and demonstrated for garages and whole bridge structures.

Design rationale: capturing of reasoning behind design decisions.

Contents

1. Introduction

Central topic of this deliverable is the usage of parametric modelling techniques for designing energy efficient buildings. The main feature of a parametric building model is that the geometry of the different building components – at least partly – is not defined explicitly (see chapter 2). Instead of this, the geometry is defined by a set of parameters and corresponding mathematical functions, enabling the generation of the explicit geometry according to the used parameter values. Advanced parametric models contain functions which take care of geometrical dependencies between model parameters. With such a parametric model, a corresponding modelling tool is able to ensure the consistency of the complete model in case a value of a single parameter is changed. For example, when the length of a certain wall is changed, an advanced parametric modelling tool would automatically change all building components which are geometrically connected with this wall, such as windows.

A number of commercially available BIM tools support the paradigm of parametric modelling and are able to persistently store their internally used parametric model in a native format. A general weakness for STREAMER is that no existing BIM tool supports a standard for importing and exporting parametric BIM models. The IFC standard, being supported by all major BIM tools and used as central format for exchanging BIM data in STREAMER, is not able to represent advanced parametric models. The development of "Parametric IFC" has started (see chapter 2), but is still in a state where it is neither supported by any existing BIM tool nor can be used for one of the design tools prototypically developed in STREAMER WP 6. Thus, the usage of traditional parametric modelling techniques and tools in STREAMER is very limited.

Much more important for the STREAMER design process is another kind of parametric modelling: The modelling and formalized representation of requirements on the needed building design, resulting in a requirement model. In STREAMER WP6, task 6.1 "Semantic design configurator", methods and prototypic design tools are being developed:

- To generate requirement models (Brief Editor and Knowledge Editor, Deliverable D 6.1 "Configurator of workflow and building process requirements);
- To use requirement models in the early design phase to automatically generate design alternatives (Early Design Configurator (EDC), Deliverable D 6.2 "Configurator of parametric design solutions");
- To check whether an IFC based design models fulfils all requirements specified in a design model (Design Validator, also D 6.2).

The intended goal of this deliverable is to support the EDC to translate a Programme of Requirements (PoR), into a spatial arrangement of rooms and functional areas within a given building shape. This is the main function of the EDC.

The PoR is a parametric description of spatial and functional requirements of needed rooms and functional areas. The EDC demands a set of general design rules which represent the "tacit knowledge" of design experts to be able to translate a PoR into design proposals. These design rules are imported into the EDC.

For reasons mentioned above, the scope of this deliverable has been extended with the development of design rules. The deliverable will thus focus on the current state-of-art in parametric modelling, the possibilities of using parametric modelling in STREAMER, AND design rules.

The sequence of chapters in this deliverable is similar to the design process of the STREAMER EDC, as can be seen in the image below;

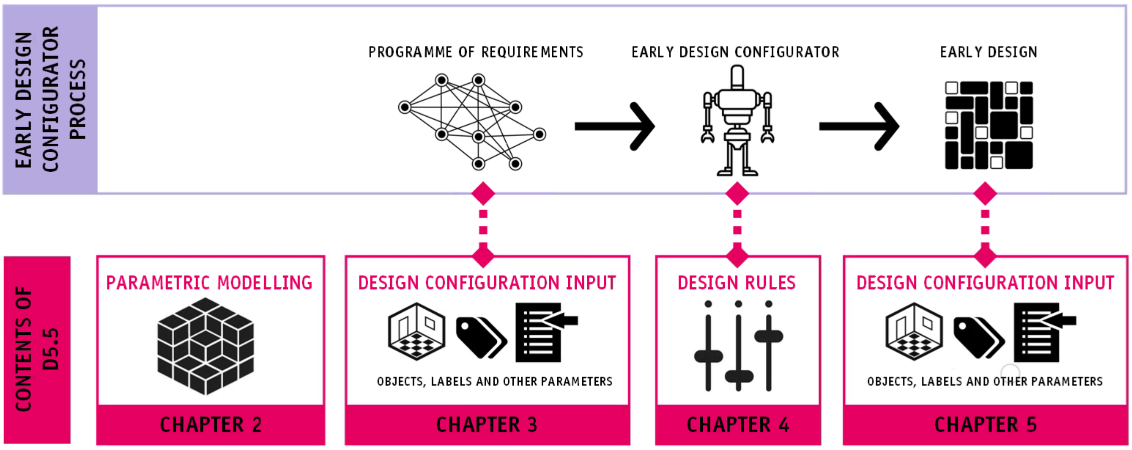


Fig. 1: graphical representation of this deliverable's content.

In the center of the image, the Programme of Requirements (PoR) is shown as the main input for the EDC. The required content for the PoR is specified in chapter 3.

The EDC uses the design rules (chapter 4) and parametric modelling techniques (chapter 5) to model the early design. The EDC is a tool to semi-automatically generate proposals which fulfill formalized requirements as best as possible. It will be embedded in a toolchain where additional tools are provided by the STREAMER partners.

Among others, those tools are generating the requirements used by the EDC and use the proposals generated by the EDC to calculate the KPIs. This validation by downstream tools requires compatibility with the early design and vice versa. This is addressed in D6.1. The content of the early design is specified in chapter 5, in a similar way as in chapter 3.

2 Parametric modelling

2.1 Current state-of-the art in parametric modelling

In this section, a quick overview of the current status of parametric modeling is provided, first on a very generic level, and then focusing on its application to CAD, AEC CAD, BIM, and finally IFC.

Parametric modelling applies mainly to:

- Geometry modelling.
- Expression of data constraints.

In the AEC sector, parametric modelling should not be limited to geometric modelling. It can be used for almost all information that constitutes a complete building model:

- Form (shape & size)
- Attributes (materials & physical properties)
- Relationship/Assembly (how parts are related, what moves with what)
- Behavior (doors opening and closing, structural members under load)

2.1.1 Parametric modelling

A very generic definition would be: “Parametric modelling uses parameters to define a model”. Examples of parameters are: dimensions used to create model geometrical representation, material, material properties, formulas to compute some values used in the model, …The main advantage of the parametric modelling is that, once the parameterization is defined, the whole model can be adapted by simply changing the values of some parameters. When one modifies a parameter, the model will update to reflect the modification.

Often, parametric modelling of geometry is completed by the notion of assembly. Several parts can be defined parametrically, and then gathered “parametrically” to create a complex object.

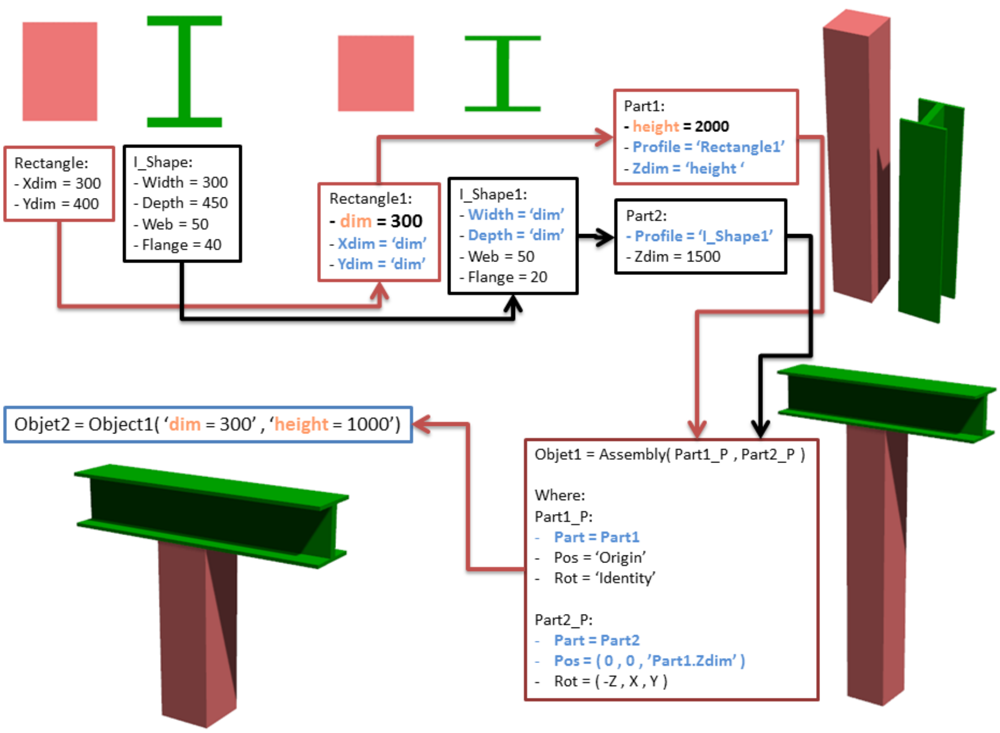


Figure 2: Example of Parametric modelling

Figure 2 describes a simple example of parametric modelling. In order to produce a parametric model of "an I-Shape beam on top of a rectangle column", the following steps are necessary:

- The first parametric elements in this example are the shape of the profiles used: instead of been described by a polygon, each profile is defined by some parameters (2 dimensions for the rectangle profile, and 2 dimensions and 2 thicknesses for the 'I shape'). We can see here the first advantage of parametric modelling: compactness of the description. Only few values are needed instead of several 2D points.
- From the 2 initial shapes, 2 other versions that "depend" on the value of a global parameter "dim" can be defined. For that, the dimensions of the 2 profiles are "related" to the value of "dim", to ensure, for example, that the 2 shapes have the same bounding-box.
- Next step is to define 3D object parts from the profiles, by defining these parts as "vertical extrusion of the profiles". A new global parameter is defined that represents the height of part1.
- Then, "part with transform" are produced by adding a translation a rotation to the parts.
- In order to build a complex object from the parts, we assemble them by applying a rotation to part2 and defining its position as: "Z altitude of part2 depends on part1 height". This way, we ensure that when modifying the height of part1, part2 will be positioned exactly on top of it.
- Finally, an occurrence of the complex object can be defined simple by giving values for its 2 parameters: "dim" and "height".

If parametric modelling brings advantages of compactness and flexibility, we can see with this very simple example that the major drawback is a much larger complexity of the design process. Even a quite simple design can lead to the manipulation of quite complex mathematical operators and a large number of parameters. Parametric modelling is very powerful, but requires more skill in model creation.

2.1.2 CAD parametric modelling

For many years now, several CAD software editors have implemented parametric modelling tools. These tools can be very useful in different contexts:

In a very early design phase, when the designer wants test very efficiently several design options. The optimized sketch is then used as a starting point for the detailed modelling of the final model (eventually also with parametric modelling).

For very sophisticated designs, using highly complex shapes, a parametric modelling process is often the only way to ensure the global consistency of the model.

When the size of the model and its complexity increase this would make it impossible to manually generate all the mechanical and industrial parts of the model. The image below shows a good example of such a project, where the general shapes of the model were optimized parametrically before generating automatically all the product components.

Figure 3: Example of parametric modelling of complex surface

Among the various CAD tools equipped with parametric modelling tools, we can mention the very powerful CATIA from Dassault Systems, which is widely used in aeronautics, aero spatial, and automotive industries.

Another very interesting software for powerful parametric modelling is Rhino when used with its free “GrassHopper” plug-in. If Rhino is very well known as a generic modelling tool (with uncommon capabilities to handle nearly every 2D and 3D file formats), the addition of GrassHopper plug-in brings a very powerful graphical parametric modelling user interface. Parameter operators are boxes that the user connects to each other to define exactly how to build a model. It is very ergonomic and efficient.

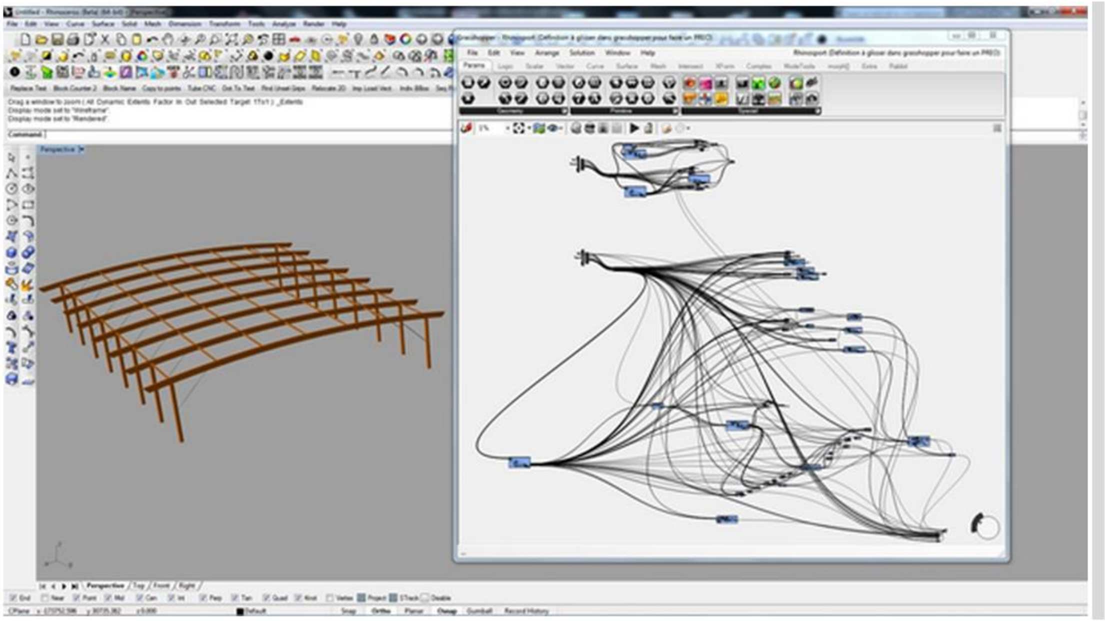


Figure 4: Rhino and its "GrassHopper" graphical parametric modelling plugin

2.1.3 BIM & parametric modelling

When it comes to the AEC sector CAD tool, several software companies have included parametric capabilities in their solutions.

Digital Project from Gehry Technologies is based on Dassault System Catia. It brings BIM semantic on top of Catia powerful modelling engine, and has proved its efficiency on several prestigious projects like the recent “Louis Vuiton Fondation” and its sail ship like design (see image below).

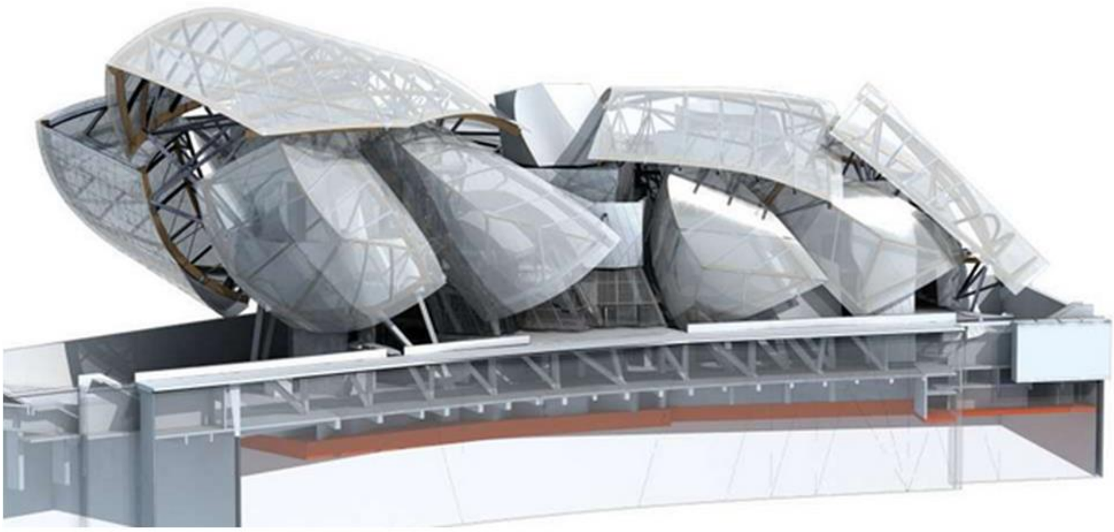


Fig. 5: one of Gehry’s designs in Catia

Revit (from Autodesk) includes some parametric functionality:

Its “family” concept allows defining parametric templates of the main entities, which can be used as libraries. The user can create its own family, for example from the generic wall family. He can define the layers of the wall, their material and thickness, and use this family to draw walls directly. Once the model is done, the user can modify parameters of the family (materials, dimensions) and the model will be automatically updated.

Some specific entities like “curtain walls” have a dedicated parametric modelling tool, allowing the user creating a curtain wall from a parametric pattern (vertical and horizontal frames).

ArchiCAD (from Advent) has similar functionalities, like parametric curtain wall tool. It has also a concept of parametric element temple creation, which can handle quite complex algorithms, using the GDL language.

Based on the solution “Rhino 3D + GrassHopper” we presented in previous section, Jon Mirtschin[1] has develop a specific plugin, named GeomtryGym, allowing to use parametric modelling capabilities of GrassHopper to generate IFC model.

[1] http://geometrygym.blogspot.fr/

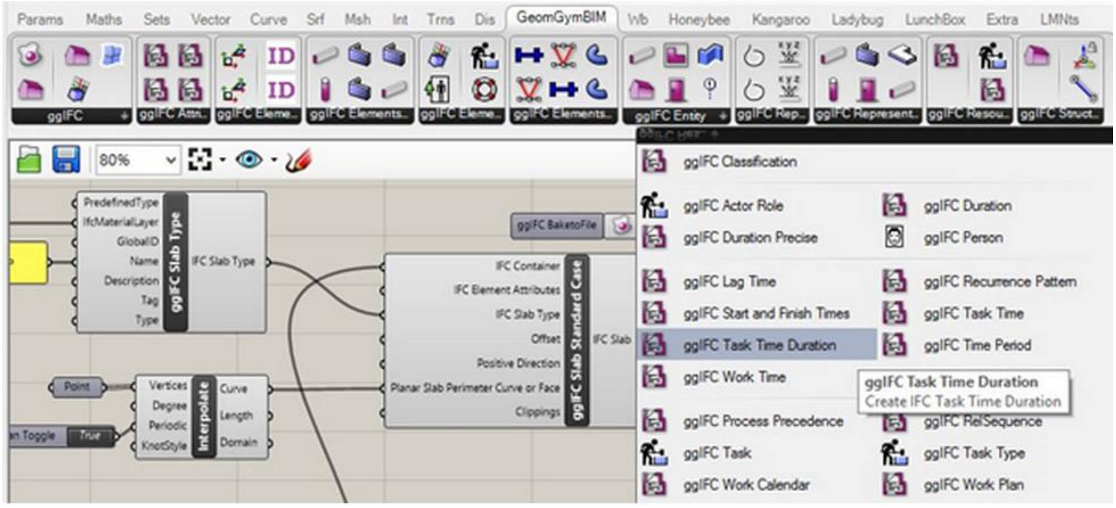


Fig. 6: screenshot of Geometry Gym

So, we've seen that parametric modelling solution for AEC sector exists. But we must admit that mainly parametric modelling is a question of tools: To efficiently produce a model using parametric design, you need an ergonomic user interface and powerful algorithms.

On the other hand, BIM is a question of Interoperability (remember BuildingSmart was once called IAI: International Agency for Interoperability…). In a BIM conception process, it is very important to exchange the model between different partner, using different tools. Today, parametric information defined in the tools we've described cannot be shared easily between software. The problem is the lack for a standard format to share the parametric information of a model. In conclusion, parametric modelling appears to be VERY important for the future of AEC sector and should be developed, but it is hardly compatible with current standards available for BIM.

2.1.4 Parametric IFC

Natively, IFC is not meant to handle parametric modelling. IFC format has been defined as an exchange format, allowing BIM information to be transmitted from one software to another with a minimal loss of information.

But the lack parametric information in the IFC format has quickly been identified as a problem, and many consider that the lack of parametric capabilities impedes the adoption of IFC format and BIM.

To address this issue, two major initiatives have been developed: A parametric extension of IFC2x3 schema and the addition in IFC4 format of some entities dedicated to parametric modelling.

<u>PA-1 Parametric IFC</u>

These parametric extensions are now a formally proposed IFC extension project at BuildingSmart.

Parametric extension defines 4 new entities allowing binding parametric behavior to attribute of an IFC object:

- IfcRelObjectEquations: Defines a relation between an object and an equation
- IfcParametricFormula: Defines inputs and outputs of the formula
- IfcParametricBinding: Points to an object, an attribute and optionally an index (if the attribute is an array). It carries four flags to control if it can be read or written to by the user and if it can be read or written to from the model.
- IfcParametricConstant: Defines a constant value

The example below shows the following parametric behavior:

- Entity #84 defines a relation between the WallType #1 and the parametric formula #86
- #86 entity invert result of formula #90 to feed entity #88.
- #90 computes the sum of #92, #93, #99, #105, #111, #117 and #123 entities.
- #88 binds the output of #86 formula to the 'ThermalConductivity' attribute of entity #124.

Once the parametric operator is applied, the entity #124 has:
IfcThermalMaterialProperties.ThermalConductivity = 1/ SUM(#92,#93,#99,#105,#111,#117,#123)

```
#1=IFCWALLTYPE('1234567890123456789012',#7,'CW','Cavity Wall',$,$,$,$,$,$,.NOTDEFINED.);
...
#84= IFCRELOBJECTEQUATIONS('U_Calculation','Calculation of overall U Value of Wall',#1,(#86));
#86= IFCPARAMETRICFORMULA('Step 1',$,(#88),(#90),.INVERT.,$);
#88= IFCPARAMETRICBINDING('U_Overall','Overall U',#124,'ThermalConductivity', $,.T.,.F.,.T.,.T.);
#90= IFCPARAMETRICFORMULA('Step 2',$,$,(#92,#93,#99,#105,#111,#117,#123),.ADD.,$);
#92= IFCPARAMETRICCONSTANT('S_0','External Surface Effect',IFCREAL(0.02),$);
...
#124= IFCTHERMALMATERIALPROPERTIES(#10,$,$,$,0.2);
```

The schema is available as an express file, and some experiments have been done to validate the concepts. But so far, very few tools are compatible with this extension, and very few file examples are available. Nevertheless this parametric extension is a very interesting initiative which must be kept going. The growing interest for BIM object libraries naturally emphasizes the need for reliable parametric model exchange format.

IFC4 Parametric Modelling

The parametric extension used with IFC2x3 is also compatible with IFC4 version.
However, some specific way of defining parametric information have been defined directly into IFC4 schema, using existing classes. This very early work consists of implementation guidelines, explaining how to create some parametric behaviors using existing entities. Apart from some slide shows and quick presentation, no real implementation and experimentation is known so far. Task 5.3 of STREAMER project could be the place to investigate further…

> "I state all the time, IFC isn't perfect. But it is capable of a lot more than implementation demonstrates at this point in time. And before we can really work on improving it, we really need to be able to test and utilize what it is already capable of to identify improvements and shortfalls"
>
> Jon Mirtschin (Geometry Gym)

2.2 Parametric modelling in STREAMER

From what has been presented in the previous chapter, it appears that design tools using parametric modelling can be very powerful project design tool, allowing:

- Creating complex models from a set of parameters,
- Making easier the evaluation of options and variants,
- Optimizing the modification process.

However, parametric modelling software can't efficiently exchange the parametric information, due to the lack of parametric exchange format. Several initiatives around the IFC format have proved that IFC (4 or 2x3 with extensions) could handle parametric behavior, but no real implementation of this feature exists in current software.

This leads us to the discussion about the relevance of parametric modelling in STREAMER project. It is needed to distinguish two complementary aspects of parametric modelling:

- Ability to build a model using parameters and constraints.
- Ability to exchange parametric models during the design process.

The first aspect would clearly be interesting in STREAMER, and already implemented in some tools. For example, the Early Design Configurator imports a "Program of Requirements", which can be seen as a set parameters, constraints and rules, and produce a sketch model. As it exports the model into IFC format, EDC enters the category of "Parametric Generator of BIM/IFC". But in the current version, it is not a "Generator of Parametric BIM/IFC", which is needed to deal with the second aspect.

One of the questions Task 5.3 has to answer is:

"How much parametric modelling information do we need to share between components / tools in the STREAMER process?"

To answer this general question, several aspects must be taken into account. The need for parametric modelling / parametric information exchange may depend on

The scale considered. In STREAMER, we consider the following scales:

- Site / Neighborhood
- Building
- Functional area
- Room / Space

The phase, and more precisely on the objective of each phase considered in STREAMER:

- Program of requirement

- Early design
- BIM / GIS integration
- Design validation
- Energy simulation

A more precise question would be:
"What parametric information do we need to handle and share at each phase and at each scale?"

Propositions for the level of parametric modelling to be included in STREAMER:

- Keep a pragmatic approach, by choosing to use "generator of IFC" when sufficient: for this purpose, tools can deal with input information (modelling it, checking it, using it), but will not share it with other tools.
- Identify some strategic Phase/Scale in which "Key parametric information" would need to be shared using IFC. In these cases, experiment implementation of IFC parametric modelling.
- Consider more advanced use of Parametric IFC for products and component systems (parametric IFC catalog model)

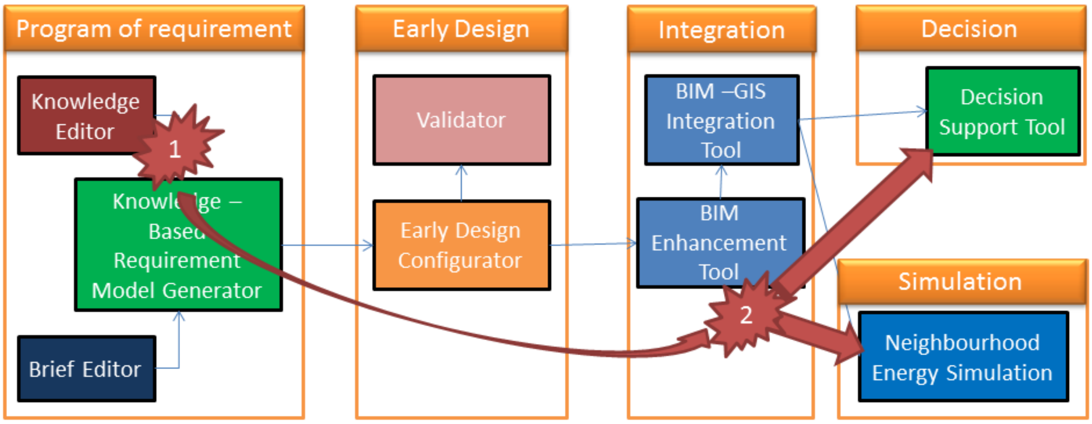


Figure 7: Synthetic representation of phases and tools. Explosion symbols stand for the presence or need for Parametric modelling capabilities of tools. Arrows identify the need of exchanging parametric information between tools

The Early Design Configurator includes modelling of the sketch BIM model. It may be interesting to extend its capacities to the creation of walls, slabs and openings in order to produce BIM envelope.
If not generated by the EDC, BIM envelop objects would need to be generated by other tools (some kind of BIM enhancement tools), to be added or developed in the project. This information is necessary to address energy simulation.

It would be interesting to evaluate the possibility to express the Program of Requirement and Design Rules in a parametric IFC format that would be imported by the EDC.

EDC may export some variants or options for systems /component used in the design. This information could be used by the decision support tool to evaluate different options or variants, or the energy simulation tool, to run several scenarios.

3 Design configuration input

In deliverable 5.2, a more detailed description of these objects and attributes will be provided and captured in Reqcap (explained in chapter 7), but because deliverable 5.2 is due one year after this deliverable, here is a summary of the most important objects and their attributes.

3.1 Objects

Building

The interface of the EDC lets the user define the boundaries of the building in which the functional areas / rooms / spaces are to be allocated. Specifying these boundaries manually is mandatory, which means that the building shape and size must be known beforehand. So although the building component itself is not necessarily contained in the PoR, its properties do belong to the input required by the EDC. As mentioned in deliverable 3.1, the deep or narrow plan configuration of a building plays a big role in energy consumption. It is therefore recommended that the designers choose deep or narrow building shapes consciously, as a starting point for the EDC. Energy analysis tools are expected to give feedback into the most energy efficient solution afterwards.

Storey

A storey is a building level, which can be referenced in the PoR. For instance, if an outpatient department back office should be on the 2nd floor to connect to a similar functional area in an adjacent building, this requirement should either be translated into a design rule specific for the project, or the design team should predefine the location of this functional area within the EDC. Technically, the storey becomes an object once created by the EDC.

Functional area

As described in deliverable 1.1, functional areas are an important ingredient of the STREAMER project. (Some examples of functional area: Day surgery, Diagnostic imaging, Mortuary). The EDC is able to place the functional areas in the building as specified by the design team before it arranges the rooms and spaces within the functional areas.

Room and space type

It is important to distinguish between programmed and non-programmed Space Units. Programmed Space Units are provided in the PoR; non-programmed Space units are not. Both must be created by the EDC. For example, an office is a Space-Unit that is mentioned in the PoR with a minimum area, but a corridor is not. Corridors are created by the design team and the area is not known until the design has been made. To be able to make a distinction, we have decided to use the ISO definitions (ISO 6707-1:2004(E) to divide Space-Units into spaces and rooms:

- Space: Area or volume bounded actually or theoretically (ISO chapter 4.1.1)

- Room (ISO chapter 4.1.3): Enclosed space (ISO chapter 4.1.1) within a storey (ISO chapter 4.1.2), other than a circulation space (ISO chapter 4.4.1).

A room type is only existent in the PoR, where it can have multiple instances. (Some examples of room type: Kitchen, Nursing Station, Office). For example, the PoR can contain 1 room type “patient room”, with 10 required instances. Once these 10 rooms have been created in a design (either manually, or by the EDC) they are simply called “rooms”. Room types can be different because of their name and properties.

Although spaces are usually not specifically described in a PoR, it is possible to describe them in a generic way. For example: “all corridors in functional area X must be at least 2.4 meters wide”, implies that this corridor is a space type. The structure of the design rules is able to accommodate these kind of requirements. Once created, either manually or by the EDC, the corridor becomes a space.

Building services component system

In Deliverable 2.2, building services component systems have been described in addition to individual components. These component systems represent a collection of coherent components, such as building services systems. Using component systems instead of individual components in the early design phase avoids introducing complexity into the BIM without adding to the quality of the design itself.

3.2 Labels and other attributes

The design rules can describe the relations between different classifications of objects. Those relations between classifications can be derived from the name or attribute of a room.

Labels

The use of semantic information in the design process is not new at all; in fact, architects are used to having semantic information to structure their designs, as shown by the following image:

Fig. 8: Representation of a PoR, as sometimes used by architects when working with physical models.

Here, rooms are represented by foam blocks which are sorted by color. The colors can be a visual representation of a semantic label. When designing the model, the blocks are manually arranged.

In deliverable 1.2, default label values have been assigned to most common room types, spaces and functional areas as defined in STREAMER. This can also be considered as a sort of "library of objects". After all, the rooms, spaces and functional areas are objects in the BIM. The library can be used to generate a PoR which is fully compatible with the EDC and the design rules. Currently, the label definitions are still under development. Therefore, the label definitions as used in this deliverable have been enclosed (Appendix 3).

A selection of other attributes that are used in the PoR:

Location (containing address, containment in site, etc.)
The EDC uses OpenStreetMap to place the building on a site, giving it a geolocation. Having this context is important for downstream analysis tools (for example, shading from surrounding buildings).

Size (required area, minimum width, height, etc.)
Geometric properties are an absolutely essential ingredient of the PoR. The EDC will generate the functional areas, rooms and spaces in compliance with these requirements.

Belongs to functional area
This attribute belongs to the room or space types, which must be assigned to a functional area. The EDC uses this attribute to apply design rules which reference a functional area.

4 Design rules

4.1 Introduction to design rules

One of the earliest common hospital types is the Asclepieion, which was founded on the principles of the Asclepieions in Greece, around the 5th Century B.C.

> An important contribution of the Asclepieions is their holistic view of the human being's endeavours and surroundings. Considering the limited medical knowledge of the time, they provided as therapeutic an environment as could be wished for. In the Asclapieions this entailed patients taking part in interactive activities, which may have involved relatives, friends, artists in residence and staff. Art was omnipresent in all its forms because it formed part of the holistic ethos that was used to promote well-being. (Kjisik, 2009)

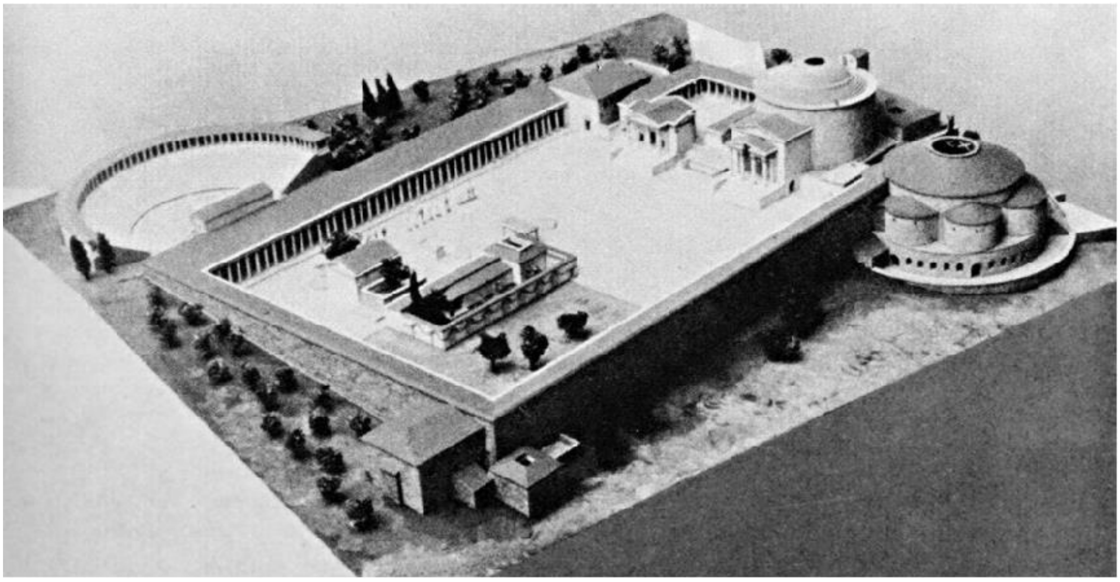


Fig. 9: Asclepieion, Pergamon (Thomson and Goulding, 1975: The Hospital: A social and Architectural History).

Recent interest in the “soft, non-medical” aspects of patient care have led to an increasing amount of evidence-based design[2] studies and a renewed interest in historic precedents like the Asclepieions.

When asked to incorporate some of these elements of the Asclepieions into modern hospital design, a contemporary design team might be inclined to literally incorporate spatial configuration elements of the building shape, for example the big square or the theatre. This type of integration is unlikely to succeed, because clearly the layout of the Asclepieions is incompatible with modern medical treatment processes, and our modern perception of art is not necessarily associated with large scale live performances.
So instead, wouldn’t it be more interesting to apply some of the design reasoning, or way of thinking, of the Asclepieons to achieve the desired result?

And if so, what is this “design reasoning”?

In this deliverable, a method is presented in which the reasoning behind design decisions is captured in so-called design rules. Design rules are used by designers to support their decisions, and reveal why a building looks a certain way instead of showing how it looks. Ideally, the design rules should be evidence based as much as possible, although that is beyond the scope of this deliverable.

[2] “Evidence-based design, or EBD, is a field of study emphasizing credible evidence to influence design. This approach has become popular in healthcare to improve patient and staff well-being, patient healing, stress reduction and safety. Evidence-based design is a relatively new field, borrowing terminology and ideas from disciplines such as environmental psychology, architecture, neuroscience and behavioral economics”. In Wikipedia. (23 June 2015). Retrieved from https://en.wikipedia.org/wiki/Evidence-based_design on 5 August 2015

4.1 Design rules and the STREAMER methodology

The STREAMER project possesses a very interesting element that adds another dimension to this deliverable in comparison to other studies into design rationale: design automation. The design rules are intended to be used by the EDC. The EDC creates an optimized spatial configuration of required spaces by using design rules. The design rules can be prioritized by providing KPI settings within the EDC, allowing the creation of multiple designs, all using the same design rules.

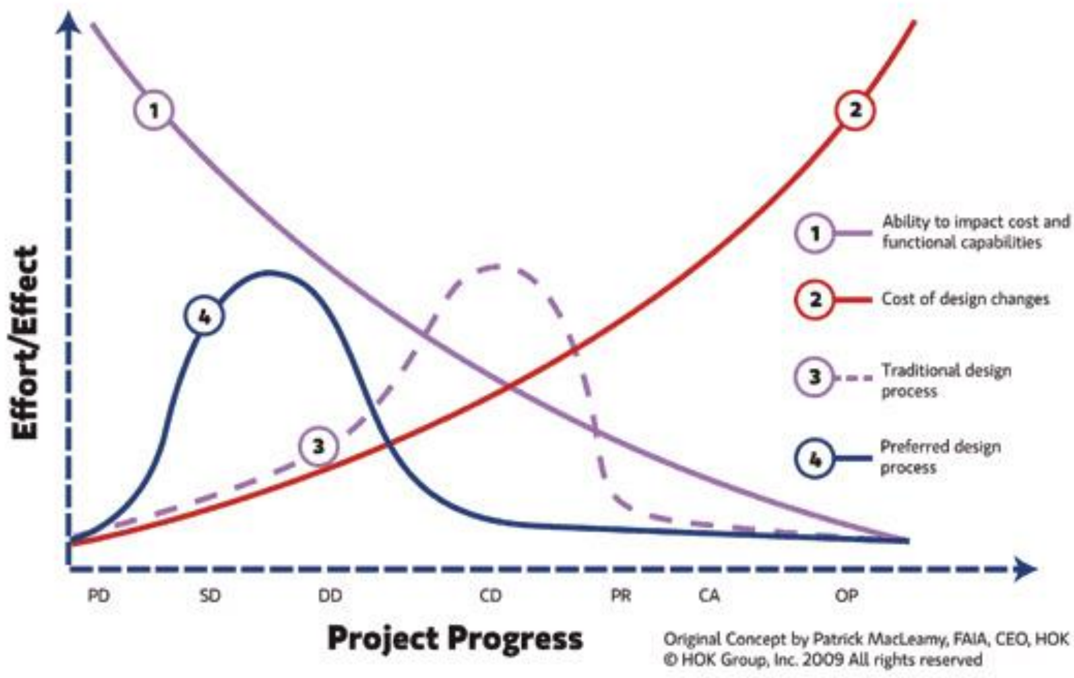


Fig. 10: Progress/effect graph.

In the STREAMER DOW, this image plays an important role. In the envisioned process, the design effort (line #4) peaks in the early design, where it has a high impact on the design performance (line #1) at the lowest cost (line#2). The availability of design rules and the expert knowledge that it represents is expected to help designers to make design decisions from a well-informed position and within a limited amount of time in this early design phase.

Design rules are used by the design team to support design decisions. Design rules as used by architects are based on many factors which include expert knowledge, personal preferences, personal experiences, conventional wisdom, regulations, cultural context and medical requirements. As architects become more experienced professionally, personal preferences and experiences lead to internalized, personalized design rules; even architects collaborating on the same project can have different design rules in mind. Needless to say, architects do not always agree on which design solution works best…

Although design rules contained in building regulation can easily be found, the documentation of design rules by designers seldom occurs. Due to the dependency on STREAMER-specific attributes to which the design rules are applied, the design rules in this deliverable have been created from scratch.

Due to the context-specific nature of the factors underlying the design rules, it is neither achievable nor desirable to produce an optimum or complete set of design rules that can be used in every situation. Therefore the primary focus has been on general design rules which are not too project-, person- or location-specific and with relevance in the early design phase.

In a design process, multiple design rules are usually applicable at the same time; the challenge for the designers is to find the design solution which offers the best trade-off between these design rules.
Because of the complex relation between design rules and final result, reconstructing design rationale by people who have not been involved in the design process is almost impossible. For example, hospitals can have different configurations of 1-/2-/3- or 4-person patient rooms. Has the decision for a certain configuration been inspired by regulations, a contract with a health care insurance company, cost reduction, ideas about patient wellness, compatibility with the structural grid size, the staff's willingness to cover a certain distance or the personal preferences of the hospital management? Although reading through the minutes of design meetings might provide some clues into the considerations behind certain design choices, most likely it was a mix of documented and non-documented influences.

Although the design rules have been created as input for the EDC, their presentation in natural language allows them to also be used in a manual (human) process, for example to validate an existing design. The design team can use this general, predefined set of design rules in a format that allows changes or additions to be easily made for a specific situation. In the design rules developed in this deliverable, non-negotiable rules have been clearly identified, although they can be modified. For example, the maximum evacuation length is part of building regulation, although the maximum length varies between countries.

Because the EDC does not use favoritism like humans do, some interesting / surprising results might be expected, although at the time of writing it's too early to tell whether the quality of these results is good enough to be seriously considered in the design process.
However, developments in other fields indicate that application of tools like the EDC can be of added value to the design. Ship designers in The Royal Netherlands Navy, for example, use a design methodology called the "packing approach". Like the EDC, this approach is also based on design rules and a PoR.

The following image shows several examples of early ship configurations as created by the “packing approach”:

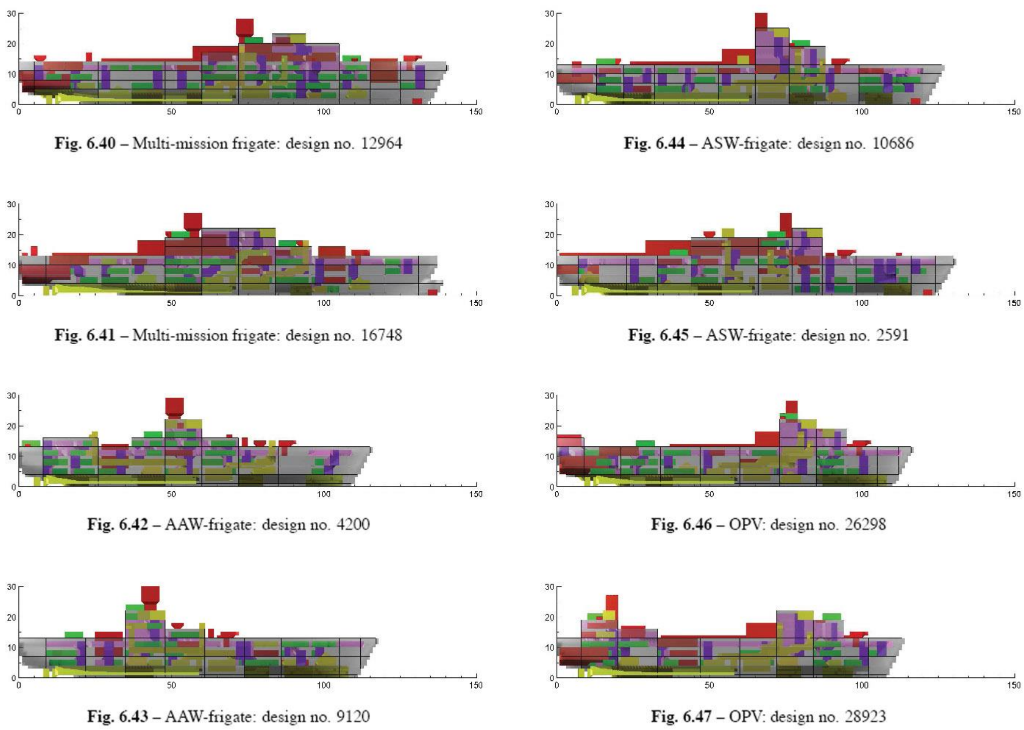


Fig. 11: Generated results by the “packing approach”, showing two early ship design configurations for four ship types (2011, Van Oers)

As can be seen by looking at the design numbers mentioned below the results in figure 10, computers are able to create huge amounts of design alternatives. This immediately creates a problem, because who is going to review all these alternatives and select the most suitable designs? Clearly, when designers generate this many design alternatives, they need to be supported by validation tools. In STREAMER deliverable 3.6, a decision-support tool will be developed in which the ‘performance’ of these design alternatives is displayed on a dashboard, based on the STREAMER KPIs.

A popular misconception is that the application of design rules in the context of automated design will automatically lead to less design freedom and a more “standard” outcome. However, the opposite is probably true: depending on the quality of the design rules and the way these are prioritized, the designers simply might have a wider range of options to choose from than ever before.
Another advantage is that working with design rules will make it easier to share expertise within the design team and design-related organizations, which will improve the consistency and documentation of the design, especially when multiple designers are involved.

As can be expected, all this will come at a cost, because not all design rules can be valid in every situation. Some design rules have to be tailor made per project, which will take a considerable amount of time, even for experts. And, because the semantic labels are so important for the design rules, any user of the design rules will have to be familiarized with the labels as well.

4.2 Design rule content

Corresponding with the structure of the STREAMER project, there are three types of design rules: related to spatial objects (room type / space type / functional area type / storey), building envelope solutions and MEP systems.

4.2.1 Design rules related to spatial objects

Design rules related to these types have been collected in a sheet, and should be read horizontally from left to right (Appendix 1). The design rules either relate two different spatial objects (one to one), or specify a form of clustering (one to many). In total, there are 37 columns which belong to 7 groups (from left to right):

1. Name
    a. Rule name: The rule can be identified by its name, which helps to quickly tell what the rule is about without reading it completely.
2. Context: Having information about the context may not be important for the EDC, but it is important for the design team to know why a rule is important (or not), because for every project, rules must be prioritized using the KPIs (explained later).
    a. Assumption: Subjective motivation of why the design rule is included.
    b. Argument: Description of what the rule contributes to the design.
3. Object A: First object to which the rule is applicable.
    a. Quantity: The amount of object A's to be controlled by the rule.
    b. Type of object: Must be one of the objects mentioned in chapter 3.
    c. Attribute 1: The first attribute of object that is referenced by the rule.
        i. Comparison operator: Determines the relation between the label value and attribute 1.
        ii. Label value: Should match one of the predefined values in appendix 3.
        iii. Functional area name.
        iv. Room name.
        v. Space name.
        vi. Storey name.
    d. Attribute 2: The second attribute of object that is referenced by the rule.
        i. Comparison operator: Determines the relation between the label value and attribute 1.
        ii. Label value: Should match one of the predefined values in appendix 3.
        iii. Functional area name.
        iv. Room name.
        v. Space name.
        vi. Storey name.

4. Relation: The envisioned relation between object A and object B (relation to be created by the EDC).
    a. Spatial relation: Either (in the case of clustering) the relation between different instances of object A, or the relation between instances of object A and object B.
    b. Amount: the amount of distance between instances of object A and B
    c. Distance (m): numeric value of the distance.
5. Object B: Second object to which the rule is applicable.
    a. Type of object: Must be one of the objects mentioned in chapter 3.
    b. Attribute 1: The first attribute of object that is referenced by the rule.
        i. Comparison operator: Determines the relation between the label value and attribute 1.
        ii. Label value: Should match one of the predefined values in appendix 3.
        iii. Functional area name.
        iv. Room name.
        v. Space name.
        vi. Storey name.
        vii. Numeric value: Field for the specification of a numeric value when required by an attribute such as height.
6. Argument:
    a. Argument type: specifies whether a design rule is negotiable or not. The relation between non-negotiable and the KPIs is not relevant, because these rules must always be met anyway.
7. KPI: Relation of the design rule to the KPIs as specified in STREAMER D3.2.
    a. Energy.
    b. Finance.
    c. Quality.

The sheet uses data validation to ensure consistency of data and avoid misspelling. This is essential for readability by the EDC. For example, input for the quantity column (number 3 in the list just mentioned) can be one of three options: “all”, “one” or “at least one”. The EDC is designed to understand exactly what these quantities mean. If someone would write a rule in which the quantity would be “some”, the EDC would not be able to process the rule. All predefined values are enclosed appendix 2. The column numbers of the sheet match with the design rule document, so it’s easy to see which values belong to which design rule column.

For every design, the use of design rules should be checked and if necessary changed. For example, regulations can be different for designs and thus require modification of design rules.

The design rules mention functional area and rooms by their type name. A PoR that has been created for a hospital will almost surely contain functional areas and rooms with different names than the ones predefined in STREAMER (deliverable 1.5). Either because the functional area or room is very specific (only the most common functional areas have been predefined), or it is simply called different. In this case, the design team has two options. Either the projects' functional areas or rooms should be mapped to the predefined ones, or the design rules should be modified to accommodate different names. To preserve the consistency of the design rules, the first option is preferred.

The relation between the KPIs and the design rules is very important for the EDC. In the EDC, the design team is expected to control which KPIs should be fulfilled with the highest priority, which will lead to a specific design output by the EDC. Even with a limited amount of design rules, many different design outputs can be generated with different KPI priorities. And when design rules are conflicting, the KPIs can determine which design rule has the highest priority.

Due to the possible occurrence of contradicting design rules, it is unlikely that all rules can be fully observed simultaneously. This situation will especially occur when rules concentrate on single, conflicting Key Performance Indicators of the design (e.g. energy efficiency, cost efficiency or operational efficiency, see Deliverable D 3.1 "Building-oriented EeB KPIs of newly designed and retrofitted buildings"). By varying the priority of these design rules, the EDC in its final version will be able to generate early design alternatives with a user-defined weighting of KPIs.

After the final review process, the design rules in Excel format will be available for download at the STREAMER sharepoint (Folder: Shared documents -> 01 Deliverables Final -> D5.5 design rules_150828.xlsx)

4.2.2 Design rules related to Building envelope solutions

In D1.3, the conclusion was drawn that most building envelope solutions have no relation with the activities inside the hospital (represented by the labels). So far, only a couple of relations have been identified, such as: windows vs. daylight requirement, or natural ventilation system vs. the presence of vents.

At the moment of writing, the EDC is not intended to generate building envelope solutions. However, these relations have been anticipated in the design rules. Assuming that it is possible to make a window or a vent in the facade manually in a later design phase, functional areas, rooms and spaces requiring daylight or natural ventilation are simply placed at a distance of 0 to the edge of the building mass, where the facade will be. This is a general design rule (not project specific).

4.2.3 Design rules related to HVAC component systems

An envisioned additional functionality of the EDC is the addition of HVAC component system suggestions to functional areas and rooms. (The concept of HVAC component systems is explained in paragraph 3.1). These suggestions are made by using a method similar to the one first introduced in D1.3. In short: by comparing label values of functional areas or rooms to label values of HVAC component systems, a selection of compatible HVAC systems can be made for each functional area or room. At this moment, the EDC does not support this feature, although it is expected that it can be incorporated based on the method described in this paragraph. At this moment, only the most energy efficient HVAC systems are prioritized. When finance and quality KPI performance of the HVAC systems are also known, it will be possible to integrate the systems choice with these KPIs as well. Depending on the progress with the EDC and integration with WP2, this can be researched in D5.6

There are too many combinations of different HVAC systems with different components to describe these all. For example: ‘if a room needs heating (by a water distribution system), air cooling or water cooling could be added. Heating could be provided by radiation, and cooling by an air-handling unit, or provided with a VRF-system. And because of the different possible types of ventilation concepts the number of possible solutions are huge. Also, choices for heating/cooling and ventilation are interrelated. Therefore the design rules to compare a HVAC system to room levels are on a global level and combine selection of heating/cooling and ventilation. The chosen systems are aligned with the systems in D2.2.

The HVAC systems are categorized, depending on the specific label values for hygienic, equipment, user profile, comfort, accessibility and construction class. By comparing the label values of functional areas or rooms to the capabilities of the HVAC systems, a selection of compatible HVAC systems can be made. For each HVAC system the specific energy efficiency for heating, cooling and ventilation is given. The most energy efficient system can be selected.

Depending on the relevant KPI (for example energy efficiency or comfort class) the decision maker is supported by a selection of suggested compatible systems, by excluding the incompatible HVAC systems because of mismatching with the room labels. As the energy efficiency of these systems is known, the EDC can select the most energy efficient system. The comfort rating is based on the hygienic and comfort labels.
The available component systems and their label values are coming from STREAMER Task 2.1.

As with the object related rules, the rules for HVAC component systems can be manually interpreted and therefore used in the STREAMER project independently of the EDC.

The tables below shows the required information of the HVAC component systems (here, HVAC system) to allow mapping to the functional areas and rooms.

Simultaneously a selection of systems for heating/cooling (HC) and ventilation systems can be made.

System category heating/cooling			HEATING efficiency		COOLING efficiency	Supported label values					
name	distribution (heating)	distribution (cooling)	ave;avg ≤ 50 °C	ave;avg > 50 °C	Ave;avg	Hygienic class	Equipment	User profile	Comfort class	Accessibility	Construction
HC 0	-	-	-	-	-	All	= EQ1	All	All	All	All
HC 1	Water or water and air	-	100%	95%	-	< = H4	= EQ1	All	< = Ct3	All	C1-C6
HC 2	Water or water and air	Water	90%	85%	94%	< = H4	< = EQ5	All	< = Ct3	All	C1-C6
HC 3	Water or water and air	air	95%	90%	94%	< = H4	> = EQ5	All	> = Ct4	All	C1-C6
HC 4	Water or water and air	water and air	100%	95%	93%	< = H4	> = EQ5	All	> = Ct4	All	C1-C6
HC 5	Air	-	95%	95%	-	> = H5	= EQ1	All	< = Ct3	All	C1-C6
HC 6	Air	Water	95%	95%	94%	> = H5	> = EQ5	All	< = Ct5	All	C2
HC 7	Air	air	95%	90%	99%	> = H5	< = EQ4	All	> = Ct4	All	C1-C6
HC 8	Air	water and air	100%	100%	93%	> = H5	> = EQ5	All	> = Ct4	All	C2

Fig. 12: table showing the information required for suggesting heating/cooling systems.

Ventilation system		Csys.vent.med ;nd	Supported label values					
Type	Description	[W.h/m3]	Hygienic class	Equipment	User profile	Comfort class	Accessibility	Construction
A	Natural ventilation	0.00	H1	= EQ1	All	Ct1, Ct2	All	All
B	Mechanical supply and natural exhaust	0.33	H1, H2	= EQ1	All	Ct1, Ct2	All	All
C	Mechanical exhaust and natural supply	0.33	H2	EQ1 - EQ4	All	Ct3 - Ct5	All	All
D	Mechanical supply and exhaust	0.83	> = H3	> = EQ 2	All	> = Ct3	All	All

Fig. 13: table showing the required information for suggesting ventilation systems.

An example to clarify this method: If a room/department label is H5, EQ1 and CT3, the suggested HVAC category is HC 5 (fig. 13), the suggested ventilation type is D (fig. 14).
The next step is to combine this ventilation system and heating/cooling system.

System category HVAC		Ventilation system	Building level		Roomlevel	
name	description	Type	Heating	Cooling	Heating	Cooling
HC 1	Decentral ventilation on room level	A /B	-	-	Yes	No
HC 2	Decentral ventilation on room level	A /B			Yes	Yes
HC 1	Decentral ventilation on room level	C/D	-	-	Yes	No
HC 2	Decentral ventilation on room level	C/D			Yes	Yes
HC 5	Central ventilation on Building level	C/D	Yes	No	No	No
HC 6	Central ventilation on Building level	C/D	Yes	No	Yes	Yes
HC 1	Central ventilation on Building level	C/D	Yes	No	Yes	No
HC 2	Central ventilation on Building level	C/D	Yes	No	Yes	Yes
HC 7	Central ventilation on Building level	C/D	Yes	Yes	No	No
HC 8	Central ventilation on Building level	C/D	Yes	Yes	Yes	Yes
HC 3	Central ventilation on Building level	C/D	Yes	Yes	Yes	No
HC 4	Central ventilation on Building level	C/D	Yes	Yes	Yes	Yes

Fig. 14: table showing the selection scheme of the category.

HC 5 and ventilation type D gives the typology of the HVAC system. Central mechanical ventilation on a building level, including centralized air heating system on building level, without cooling. It is up to the designer to choose between a the underlying configuration of the HVAC system based on constant flow, or variable flow air heating system. That's out of the STREAMER project.

The HVAC category has a strong relation with D3.2 for calculating the KPI Energy. To explain, again continue with the example: the efficiency of the distributing system of heating (for the selected case HC 5) will be 95% and the ventilation (Vent D) characterized energy usage is 0.83 W/h/m3 air supply. That means to calculate the energy usage the heating demand of the room will be multiplied by 95% and the quantity of the air supply to the room will be multiplied by 0.83 W/h/m3 air supply.

The parametric information HVAC 5 and ventilation type D should now be added to the room labels. That's necessary to provide the further design process with the chosen HVAC system.

4.2.4 Design rule ownership

Design rules contain a lot of expert knowledge, of which some might be considered sensitive or confidential. For example, an architect might want to protect his design rules (intellectual property) out of fear that another architect might be able to copy his or her approach to hospital design. There might also be a fear of having wrong assumptions (which can be easier identified from reading design rules than reading a floor plan) out in the open. Forcing a design team to share their design rules will make them feel exposed.

4.3 Input formats for the EDC

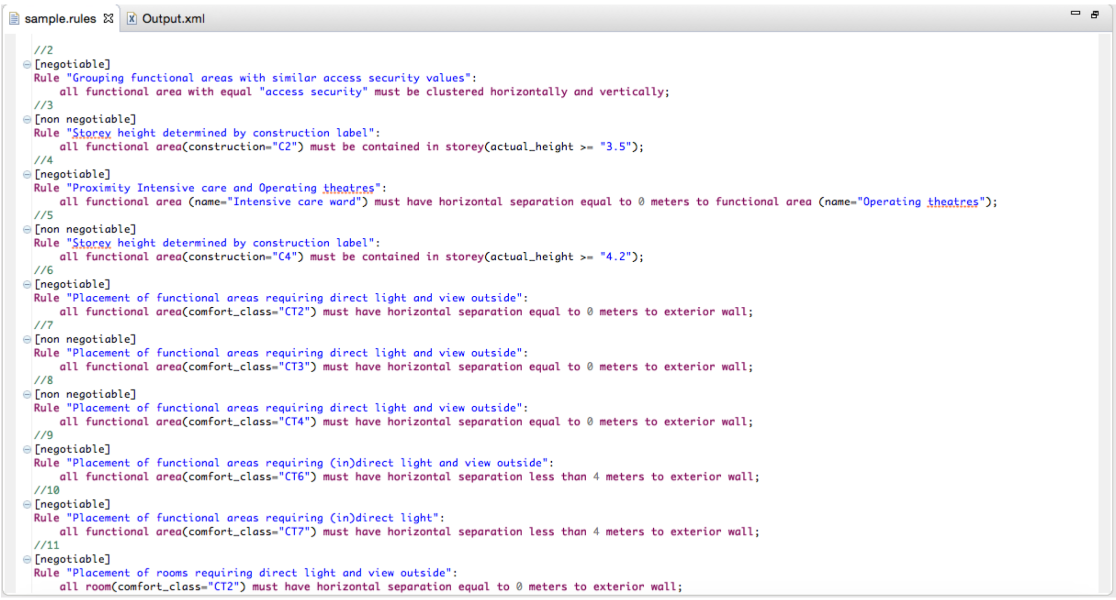
Both the PoR and the design rules are imported into the EDC as requirements.

The PoR can be defined in commercial software like BriefBuilder or dRofus and then exported as Microsoft Excel files. From Excel the data can be exported as a CSV file readable by the EDC. Another possibility for future development of the EDC is to use the IFC data format instead of the CSV file format for which no standard exists.

Because design rules are part of the knowledge domain of the design team (which is usually not familiar with computer language), the design rules have been captured in natural language. This allows project-specific modifications or additions to the design rules to be made by the designers themselves, and ensures the usage of design rules is not entirely dependent on the EDC. The design rules are easy to read and can be used in a manual design process as well. The design rules were created in sheets (similar to Excel) and are enclosed as Appendix 1.

The Knowledge Editor (developed in D6.1) provides a specific formal language (so called Domain Specific Language" of DSL) to represent rules, an a specialized editor to generate DSL-rules. This language is similar to your rule formulation in structured natural language, but not equal. The user of the Knowledge Editor informally uses rule formulations to generate new representations in DSL. Only the transformation DSL → XML is done automatically.

More information about the knowledge editor is provided in deliverable D6.1.


sample.rules
Output.xml

```
//2
[negotiable]
Rule "Grouping functional areas with similar access security values":
    all functional area with equal "access security" must be clustered horizontally and vertically;
//3
[non negotiable]
Rule "Storey height determined by construction label":
    all functional area(construction="C2") must be contained in storey(actual_height >= "3.5");
//4
[negotiable]
Rule "Proximity Intensive care and Operating theatres":
    all functional area (name="Intensive care ward") must have horizontal separation equal to 0 meters to functional area (name="Operating theatres");
//5
[non negotiable]
Rule "Storey height determined by construction label":
    all functional area(construction="C4") must be contained in storey(actual_height >= "4.2");
//6
[negotiable]
Rule "Placement of functional areas requiring direct light and view outside":
    all functional area(comfort_class="CT2") must have horizontal separation equal to 0 meters to exterior wall;
//7
[non negotiable]
Rule "Placement of functional areas requiring direct light and view outside":
    all functional area(comfort_class="CT3") must have horizontal separation equal to 0 meters to exterior wall;
//8
[non negotiable]
Rule "Placement of functional areas requiring direct light and view outside":
    all functional area(comfort_class="CT4") must have horizontal separation equal to 0 meters to exterior wall;
//9
[negotiable]
Rule "Placement of functional areas requiring (in)direct light and view outside":
    all functional area(comfort_class="CT6") must have horizontal separation less than 4 meters to exterior wall;
//10
[negotiable]
Rule "Placement of functional areas requiring (in)direct light":
    all functional area(comfort_class="CT7") must have horizontal separation less than 4 meters to exterior wall;
//11
[negotiable]
Rule "Placement of rooms requiring direct light and view outside":
    all room(comfort_class="CT2") must have horizontal separation equal to 0 meters to exterior wall;
```

Fig. 15: Screenshot of the Knowledge editor.

The complete requirement model used by the EDC is generated during the import of Knowledge and the PoR by linking the PoR with the supplied knowledge as shown in Figure 16. Objects are linked by deriving the defined room types in the brief requirements from the classifications in the knowledge. The classifications are either directly referenced as rooms (e.g. a patient room), or generated from expression referencing label values (e.g. a room where construction = C6).

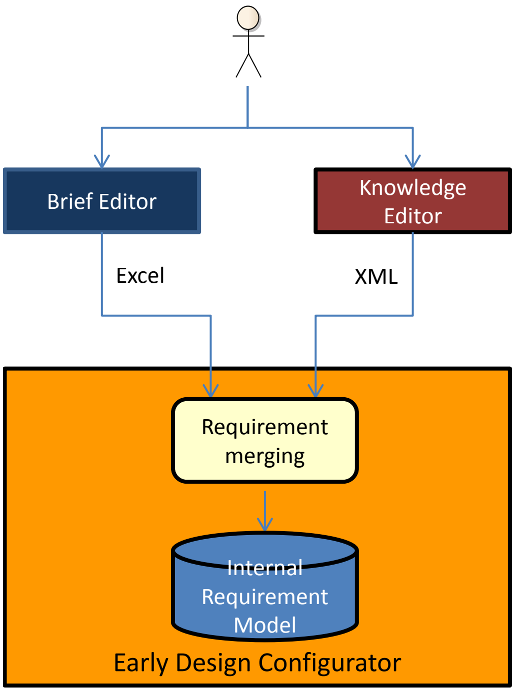


Fig. 16: Information flow scheme.

5 Design configuration output

The output of the EDC is an IFC file containing a building mass. The building is generated from the defined requirements from paragraph 3.1 and 3.2. Another future export format may be gbXML for energy estimation.

The IFC structure is an IfcProject[3] with an IfcSite. The site contains the IfcBuilding which in turn contains one or multiple IfcBuildingStoreys. The storey contains one or multiple IfcSpaces. IfcSite contains the geolocation of the building. The defined building is represented as an IfcBuilding, which contains no geometry. The geometry is contained in the IfcBuildingStorey elements inside the IfcBuilding. Each geometry is an extrusion of the 2D representation generated in the EDC. IfcSpaces are used to represent instances of the room and space types. Every IfcSpace has an extruded geometry generated from the 2D representation of the EDC. The defined functional areas of the room types are represented as IfcZones referencing the IfcSpaces. A proposal contains the floor plan of the complete building, consisting of the outline of all stories and the layout of the rooms. Those proposals can be rated and the most promising proposals are developed further.

Other attributes not explicitly in this text are added as meta data in form of an IfcProperty to their respective IFC objects. Those attributes are defined in ReqCap.

The 'component system suggestion' is an attribute attached to a functional area (Ifc Zone) and/or room (Ifc Space).

6 Relations to developments in STREAMER

6.1 Data and data exchange requirements in ReqCap

The requirements management tool (ReqCap) is used to specify data exchange requirements. It helps to collect and structure the expectations for content that is of interest in various STREAMER scenarios. In other words: using RecCap can improve the consistency of information in the STREAMER project.

One of these scenarios is to use the EDC to transform design requirements to a draft building layout. This transformation process and the design rules it uses are the main focus of this deliverable. The ReqCap tool is used to define the interfaces both to and from the EDC, which allows us to be clear about how to make use of the EDC. This is done on two levels: (1) on conceptual level using the terms of domain experts and (2) on technical level defining the link to the IFC data structure.

7.1.1 Selecting and sorting exchange requirements on conceptual level

The starting point for the work with ReqCap is to identify relevant types of data that are either directly evaluated by the EDC as input or should be attached to generated objects as output in order to be

[3] See http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcproductextension/lexical/ifcbuilding.htm and the respective other pages for other IFC elements

available in subsequent processes. They have been selected from the labels defined in WP1 or, if missing, defined as new data types. These data types then have been linked to relevant (asset) object types, which act as data containers for required information.

The described process is shown in the figure below. After adding all semantic labels and relevant object types to the ReqCap database as template definitions (tree on the left side of the figure) the requirements tree on the right side was specified by dragging and dropping elements from the left side to the right side. In the shown example the semantic label “Accessibility” is linked to “Room (type)” so that this label becomes expected information for room type definitions.

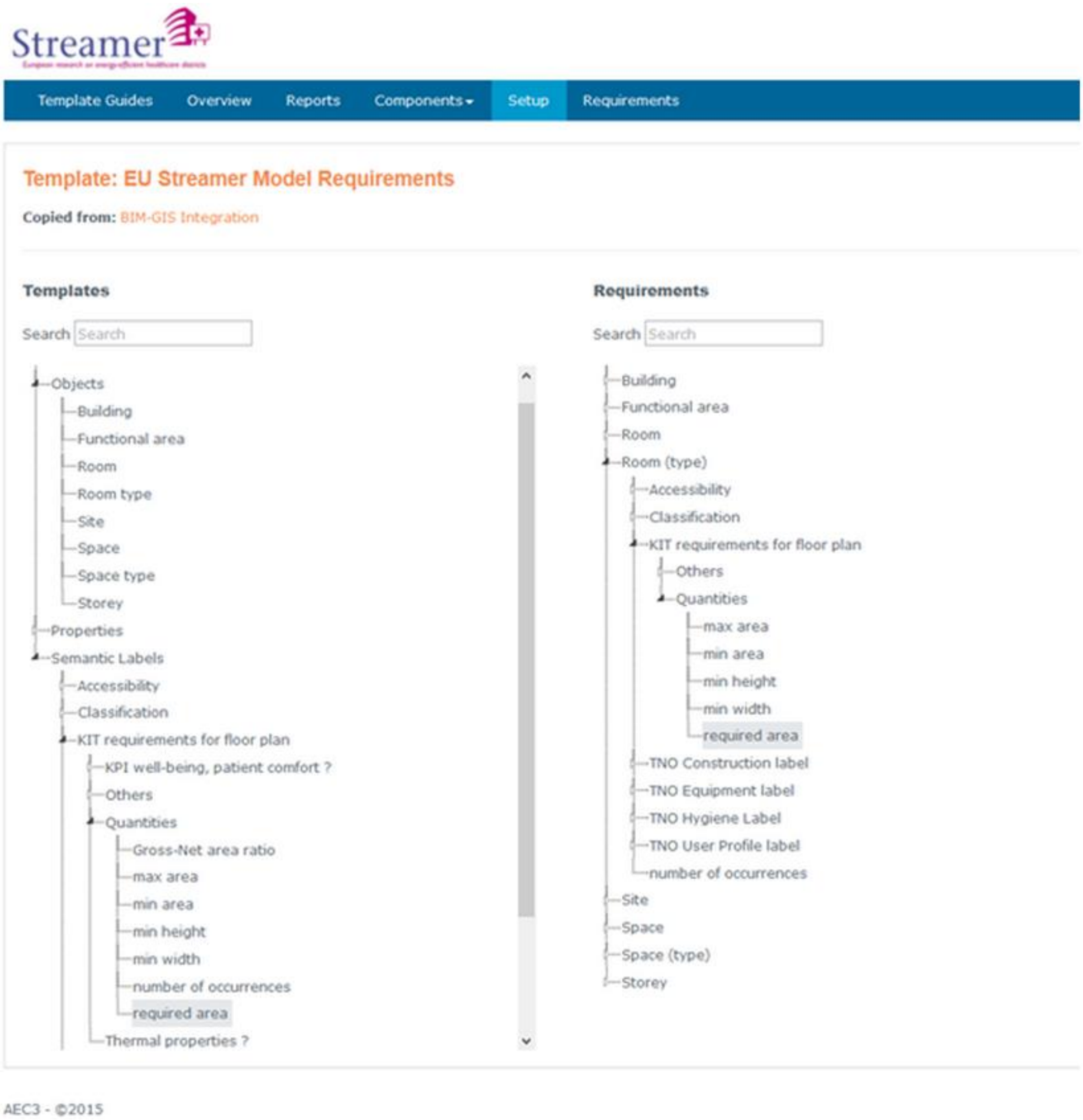
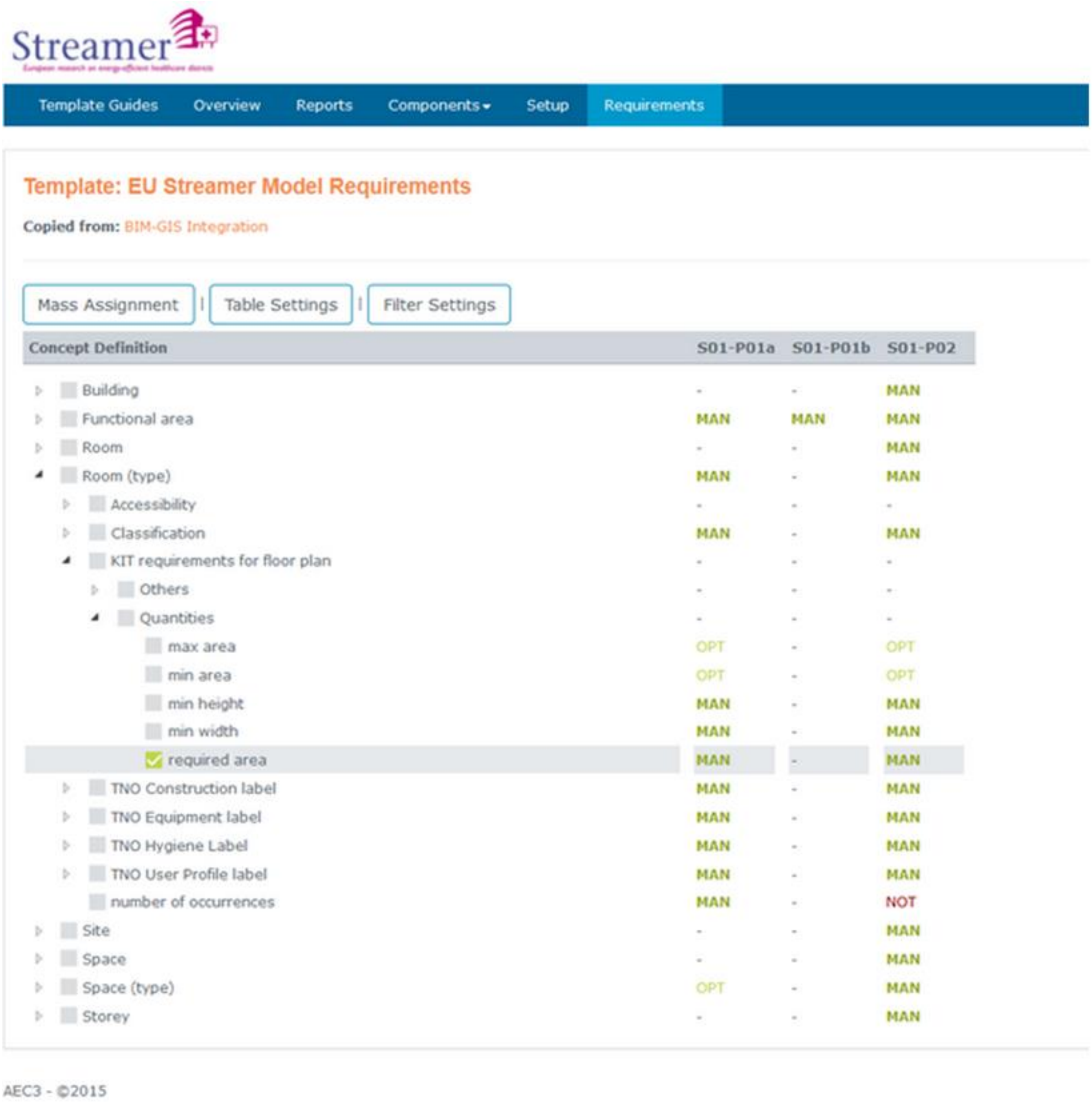


Figure 17: Screenshot of ReqCap showing the configuration of requirements by linking semantic labels to relevant object types. Note: the input has not yet been synchronized with the latest version of the labels.

After the setup of these requirements, which can be seen as an agreement how to organize and communicate data exchange requirements between domain experts, the next step was to indicate what types of data must be imported and exported by the EDC. For this, three exchange requirements have been defined; two for data import and one for data export. Throughout the discussion with domain experts it became clear that there are different scenarios in terms of how to capture requirements. The first and main scenario is to expect definitions on room type level. The second scenario is less detailed as it expects requirement definitions on the level of functional areas (zones) only. Differences between both scenarios seem to be not very big, but it means to work with a different set of design rules that must be specified and supported by the EDC.



Concept Definition	S01-P01a	S01-P01b	S01-P02
Building	-	-	MAN
Functional area	MAN	MAN	MAN
Room	-	-	MAN
Room (type)	MAN	-	MAN
Accessibility	-	-	-
Classification	MAN	-	MAN
KIT requirements for floor plan	-	-	-
Others	-	-	-
Quantities	-	-	-
max area	OPT	-	OPT
min area	OPT	-	OPT
min height	MAN	-	MAN
min width	MAN	-	MAN
required area	MAN	-	MAN
TNO Construction label	MAN	-	MAN
TNO Equipment label	MAN	-	MAN
TNO Hygiene Label	MAN	-	MAN
TNO User Profile label	MAN	-	MAN
number of occurrences	MAN	-	NOT
Site	-	-	MAN
Space	-	-	MAN
Space (type)	OPT	-	MAN
Storey	-	-	MAN

Fig. 18: Data exchange matrix specifying data requirements for different data exchange scenarios. The naming of the scenarios (currently named S01-P01a etc.) is to be adopted in the STREAMER project.

For each exchange requirement and each data type a decision is made whether the data is mandatory, optional or not required. All mandatory data defines the minimum data exchange for the covered scenario. For instance "min height", "min width" and "required area" are defined as mandatory for room types, whereas "max area" and "min area" are defined as optional as they could be covered by default values or additional design rules.

The matrix in Figure 2 shows the requirements as defined by domain experts. It can be seen as a checklist that enables to decide whether the EDC will receive or produce all expected data. If mandatory input data is missing the responsible actor is in charge to complete his task, in this case to capture the needs of the client, before starting the design configuration process.

7.1.2 Specifying the mapping to IFC on technical level

The process of defining requirements need to be completed by specifying how to hand-over that information. It is a purely technical discussion that starts with the decision about a proper data format being able to transfer requested data. For STREAMER the decision is to use the open BIM standard IFC as far as possible.

After selecting the data format it must be defined how to map domain requirements to this data format, in our case IFC. This is necessary to avoid misunderstandings on a technical level when importing the data into the EDC or exporting it out of the EDC. For the IFC data structure, the matrix can be transferred in a Model View Definition (MVD). An MVD is encoded in the mvdXML format and can be used to automate model checking as described above.

As shown in this section the ReqCap tool is being used to support activities within Task 5.3. It helped to reuse, combine and enrich specifications from other work packages, mainly the semantic labels developed in WP1. An additional benefit can be achieved if produced specifications are used for model checking, which is to be developed in Task 5.2 of this work package.

6.2 This deliverable can provide input for:

<u>Deliverable 1.2: Semantic typology model of existing buildings and districts / Deliverable 1.4: Multi-scale and multi-stage scenarios for energy-efficiency retrofitting</u>

The methodology behind the EDC and the design rules can be an important link between the identification of functional problems in D1.2 and the retrofit solutions to be developed in D1.4. The design rules developed in D5.5, can be manually (without the EDC) used in a retrofit situation, as long as the problems of the existing building have been mapped in a way that is compatible with the design rules (for instance, by using the semantic labels).

<u>Deliverable 1.6: Semantic baseline design model for new energy-efficient healthcare districts</u>

The semantic BIM template / design model for new build and existing projects should take into account the usability of the template by the format of the design rules.

<u>Work Package 2: EeB building technologies</u>

To be considered is whether the design rule approach and format is suitable for capturing the solutions for MEP systems, building envelope and building / neighborhood. Even when the EDC may not incorporate these solutions, capturing design rules in a natural language still allows usage by the design team.

Task 4.2: Semantic knowledge management

As mentioned in paragraph 4.1, the design rule approach and format is suitable to capture and re-use the knowledge of the design team. To be considered is whether a similar approach is suitable for capturing the "experience and tacit knowledge of the building operators and occupants", as mentioned in the DoW, task 4.2. If this is the case, formalized design rules could become a part of the PoR.

Task 4.3: EeB IPD framework

To be considered is whether design rules belonging to a project should be handed over to the building owner after completion by the design team. This could prove beneficial to the consistency of the design, when the building is later expanded or refurbished, because future design teams can better understand why the building looks like it does. However, there may be a problem with sensitive issues as mentioned in paragraph 4.3.4.

Deliverable 5.2: Semantic web based PMO

The in- and output specifications of the EDC (chapter 3 and 5) will be captured in Reqcap.

Task 5.2: Model based product lifecycle management

To be considered is whether the design rules should be managed by the PLM.

Deliverable 5.6: Framework for the open-source library of parametric design solutions

Naturally, the follow-up deliverable of T5.3 will continue where this deliverable stopped. At the moment of writing, it is hard to say in which direction the research will continue, since the EDC has not been tested yet in combination with the design rules, and is still limited in functionality. In the description of the follow/up deliverable D 5-6. object libraries are mentioned. However, the EDC will not use objects from a library but create the objects by itself.

Work on the design rules can be continued, and a more clear distinction between general rules and project-specific rules can be made as well as a connection to evidence-based design.

The HVAC systems suggestions made by the EDC can also be based on the finance and quality KPI´s if the method is expanded and more information from WP2 can be integrated.

Task 6.1: Semantic design configurator

As described extensively in this deliverable, the design rules are an important ingredient for the EDC.

The EDC could be enhanced with more sophisticated parametric information modelling and export capabilities as discussed in chapter 5. While Deliverable 6.1 is already finished, preliminary versions of EDC and Design Validator will not be available before project month 36.

Task 7.1: Demonstration project in the UK

The Rotherham hospital can be an interesting test case for the EDC, which may be programmed with the shape of an individual department (or the whole hospital) at TRH but it is unlikely that its exploration of the distribution of spaces would yield a plausible re-design. It can be expected that the criteria developed for model completeness for new-build will be equally applicable for 'renovation': the energy and running cost analysis can then be performed by the same tool on the same basis.

Regarding the mapping of MEP systems; for the two Departments (functional areas) at TRH, lists of possible system upgrades have already been prepared. The building model currently holds the current option and several possible upgrades as distinct Ifc Systems. The process may use the the data on Systems suitability from D3.1 to verify these lists. It will however need different capital cost parameters as upgrade costs are not the same as new-build costs: for example upgrading a lighting system may not involve replacing wiring.

Task 7.2: Demonstration project in the Netherlands

The Rijnstate north-east extension will be used to demonstrate the EDC. The PoR of the Rijnstate hospital will be converted so that the semantic information matches with the design rules and can be processed by the EDC. Within the same building envelope, the EDC is expected to generate a new design. It will be interesting to compare this design to the current design: will the EDC be able to generate a design that can be used? In any case, this test case will likely give some useable feedback for further development of the EDC.

Task 7.3: Demonstration project in France

Compatibility of the APHP with the design rules is to be determined.

Task 7.4: Demonstration project in Italy

At the moment of writing, IAA is comparing the functional areas and rooms in the SACS to the standard ones used in STREAMER in a similar process as the Dutch test case.

6.3 This deliverable has taken input from:

WP1: EeB typologies

Definitions of the STREAMER standard functional areas and rooms, the semantic labels, the concept of component systems.

Deliverable 2.2: Retrofitting solutions of integrated EeB solutions for MEP and energy systems

The matrixes of building services systems and their compatibility with the label values.

Deliverable 3.2: Process-oriented EeB KPIs in the operation, maintenance and (re) construction phases

The KPIs.

Task 6.1: Semantic design configurator

This concept of deliverable has been thought out in close collaboration with KIT to optimize the integration between the design rules and the design configurator.

7 Appendices