UNIVERSITY OF FERRARA
- EX LABORE FRUCTUS -

DA UNIVERSITYOFFERRARA department of architecture

UNIVERSITETI POLIS

International Doctorate in Architecture and Urban Planning

Semantic BIM Design Methodology for Energy Efficient Building

Candidate: Thorsten Lang
DA Supervisor: Prof. Roberto Di Giulio
Co-Supervisor: Prof. Ruben Paul Borg

Cycle XXIX

IUSS - Ferrara 1391

IDAUP

INTERNATIONAL DOCTORATE in ARCHITECTURE AND URBAN PLANNING

Cycle XXIX

IDAUP Coordinator Prof. Roberto Di Giulio

Thesis Title

Semantic BIM Design Methodology for Energy Efficient Building.

Curriculum Tecnologia dell'Architettura (SSD ICAR/12)

Candidate

Thorsten LANG
(Matr. N.118462)

Supervisor DA / POLIS

Prof. Roberto DI GIULIO UniFe

Co-Supervisor University Of Malta

Prof. Ruben Paul Borg

Semantic BIM Design Methodology for Energy Efficient Building

Abstract

The purpose of this thesis is to improve the early stage design process, within computer aided design, allowing more design options to be compared and anticipating feedback on choices that influence the building's energy consumption. To reach this purpose it analyses the role of the human designer and the encountered limits of the computer aided design process.

The research opens on the state of the art by investigating relevant open standards in Building Information Modelling (BIM) and semantic web, design automation in layout generation, early stage energy simulation and evidence based design.

As a practical application, the author took a layout generating tool developed within Streamer, the European applied research project he collaborated on. A semantic BIM design method using open standards for early stage design is described, the main building typology referenced is hospitals, suitable for their strict functional requirements and high energy consumption.

Within the method's workflow, the design process starts from a classification of Programme of Requirement data, which is then used to generate alternative design layouts. The generated proposals are evaluated by an evolutionary algorithm, based on pre-defined prioritised design rules. Several alternative designs can be exported as Industry Foundation Class (IFC) files, in which properties can be retained for downstream use. Optionally, to improve placement of building envelope elements & building services, an expert can create zones manually within the file. An automated energy calculation uses the semantic data contained in the IFC file. A decision support system is used to compare and evaluate the generated options. Both designs for new buildings and the use in a retrofitting scenario are considered.

To close the inquiry a speculative investigation of other building typologies attempts to understands the limits of the automated process and tools, and the creative contribution of the human designer. It seeks to describe which adaptations are required to design other building typologies based on concrete examples of contemporary design. The philosophy of the method and its strengths and limitations are discussed and topics for future research are outlined. In conclusion, the author proposes an iterated interaction between designer and brief to combine the strengths of the human designer and the semantic design method.

keywords

Energy Efficient Building, Evolutionary Design Algorithm, Building Information Modelling, Rule Based Design, Early Stage Energy Simulation, Semantics, Design Automation, Design Theory.

Title in Italian translation:

Metodologia di definizione semantica in ambiente BIM per costruire ad alta efficienza energetica.

Italian language abstract

Lo scopo di questa tesi è di migliorare il processo di progettazione preliminare, nell'ambito della progettazione assistita supportata da applicativi software, in modo da permettere che più opzioni progettuali vengano comparate e anticipando nel tempo il riscontro sulle scelte che influenzano il consumo energetico dell'edificio.

La ricerca si apre con un'analisi dello stato dell'arte, indagando i più rilevanti standard aperti nel Building Information Modelling (BIM*) e nel* Semantic Web*, nell'automazione della progettazione per la generazione di layout planimetrici, nella simulazione preliminare dei consumi energetici e nell'*Evidence Based Design*.*

Come applicazione pratica, l'autore ha utilizzato uno strumento software per la generazione di layout, sviluppato nell'ambito del progetto europeo di ricerca applicata Streamer*, a cui egli stesso ha collaborato. Viene descritto quindi un metodo di progettazione BIM semantico, che usa standard aperti per la fase preliminare della progettazione, la cui tipologia di riferimento è quella ospedaliera, adatta per le rigide esigenze funzionali e il grande fabbisogno energetico.*

All'interno della metodologia di lavoro, il processo di progettazione inizia con una classificazione di dati da un programma di requisiti funzionali, che viene quindi utilizzata per generare layout progettuali alternativi. Le proposte generate vengono valutate da un algoritmo evolutivo, basato su regole progettuali predefinite e prioritarizzate. Diverse alternative possono essere esportate come file di tipo Industry Foundation Class *(IFC), in cui le proprietà possono essere conservate per un uso derivativo.*

Opzionalmente, al fine di migliorare la giacitura degli elementi compositivi principali di un edificio ed i suoi servizi, un progettista esperto può creare delle zonizzazioni nel file in maniera manuale. Un'applicazione di simulazione automatizzata del consumo energetico dell'edificio utilizza i dati semantici contenuti nel file IFC. Un sistema comparativo a supporto delle scelte viene utilizzato per valutare le opzioni generate automaticamente. Possono venire considerate sia scenari di nuova costruzione che di ristrutturazione.

A prosecuzione della ricerca, l'autore svolge una indagine propositiva di altre tipologie di edifici da sottoporre al metodo di progettazione automatizzata, al fine di comprenderne i limiti e gli strumenti, nonché il contributo del progettista umano. Lo scopo è descrivere quali adattamenti siano necessari per la progettazione di altre tipologie, basandosi su esempi concreti di edifici contemporanei.

La filosofia, i punti di forza e debolezza del metodo di progettazione automatizzata vengono discussi e se ne ricavano spunti per futuri possibili approfondimenti di ricerca. Come conclusione l'autore propone un iter di interazione tra il progettista umano e il processo automatizzato, al fine di far convergere i punti di forza del metodo di progettazione semantica automatizzata e l'esperienza del progettista umano.

International Doctorate in Architecture and Urban Planning (IDAUP)
Is a PhD programme offered since 2014 by the University of Ferrara, Department of Architecture (IT) in consortium with Polis University (AL). The University of Malta, Faculty for the Built Environment (MT) is an associate member.

PhD candidate: Thorsten Lang
Email: t.lang.eu@gmail.com
Cell: (+39) 340 570 6445

Supervisor: Prof. Roberto Di Giulio, University of Ferrara
Co-supervisor: Prof. Ruben Paul Borg, University of Malta

Acknowledgements

The author has collaborated with the STREAMER European applied research project, run by a consortium of hospitals, consultants and applied research institutions. The use of its tools and processes has been essential for the development of this thesis.

"STREAMER is an industry-driven collaborative research project on Energy-efficient Buildings (EeB) with cases of mixed-use healthcare districts, that aims to reduce the energy use and carbon emission of new and retrofitted buildings in healthcare districts in the EU by 50% in the next 10 years. Such districts are the best real examples of neighbourhood with integrated energy system consisting of mixed building types (i.e. hospitals and clinics; offices and retails; laboratories and educational buildings; temporary care homes; rehabilitation and sport facilities).

This research project has received funding from the European Union's Seventh Framework Programme for Research and Technological Development under Demonstration under grant agreement no 608739 - FP7-2013-NMP-ENV-EeB."

Many thanks are due also to my family without which this thesis would not have been feasible.

List of acronyms and abbreviations

API: Application Programming Interface
BCF: BIM Collaboration Format
BIM: Building Information Modelling
BPMN: Business Process Modelling Notation
bSDD: buildingSMART Data Dictionary
bSI: BuildingSMART International Ltd.
CAPEX: CAPital EXpenditure
DSL: Domain Specific Language
DST: Decision Support Tool
EDC: Early Design Configurator, Streamer software prototype developed by KIT
EXPRESS-G: Graphical notation of relations in information data modelling
gbXML: Green Building Extensible Markup Language
MEP: Mechanical, Electrical and Plumbing
IDM: Information Delivery Manual
IFC: Industry Foundation Classes
IFD: International Framework for Dictionaries
ISO: International Organization for Standardization
KIT: Karlsruhe Institute of Technology
LOD: Linked Open Data (in context of Semantic Web)
LOD: Level of Development, or Level of Detail (in context of BIM)
LOI: Level of Information
MVD: Model View Definition
mvdXML: Model View Definition in machine readable XML format
OPEX: OPerational EXpenses
OWL: Web Ontology Language
PLM: Product Lifecycle Management
PoR: Programme of Requirements
SACS©: System for the Analysis of Hospital Equipment
RDF: Resource Description Framework
STEP: Standard for the Exchange of Product Model Data
TECT: TNO Energy Calculation Tool
TNO: Netherlands Organisation for Applied Scientific Research
XML: eXtensible Markup Language
XSD: XML Schema Definition

Conventions

Spelling used in this document is British English. The style aims for maximum comprehension and simplicity, which should facilitate the reading on a quite technical subject. To improve readability, it is written in the simple tense, in the present time except where there are evident reasons for a past or future time.

While the written passages of this document are typeset in Calibri,

`extracts from structured documents such as XML, OWL are typeset in Courier.`

Definitions

Ontology
Synonyms: semantic data model

"In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application. In the context of database systems, ontology can be viewed as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modelling knowledge about individuals, their attributes, and their relationships to other individuals (...), ontologies are said to be at the "semantic" level, whereas database schema are models of data at the "logical" or "physical" level. Due to their independence from lower level data models, ontologies are used for integrating heterogeneous databases, enabling interoperability among disparate systems, and specifying interfaces to independent, knowledge-based services." (Gruber2009)

Semantics
"Semantics is the study of meaning. It focuses on the relation between signifiers, like words, phrases, signs, and symbols, and what they stand for, their denotation". (Source: Wikipedia)

Semantic reasoner
"A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language, and often a description logic language. Many reasoners use first-order predicate logic to perform reasoning; inference commonly proceeds by forward chaining and backward chaining. There are also examples of probabilistic reasoners, (…)"
(Wikipedia)

List of figures

Figure 1: Shifting effect/cost/effort curve with a BIM workflow, MacLeamy (2004)......... 14
Figure 2: Example: Mapping project initiation using BPMN, from Streamer D5.1............. 18
Figure 3: A workflow process extended with triggering information, from Aalst 1998..... 19
Figure 4: Four layers of data schemas, adapted from bSI IFC2x4 documentation............ 21
Figure 4: Definition of spatial structure elements in EXPRESS-G, IFC2x3_MIG2009.......... 22
Figure 6: LoX; global development of LoD, LoI standards (M. Bolpagni)........................... 24
Figure 7: Conceptual model of the mvdXML schema, from bSI MSG 2013 25
Figure 8: MVD Concept template of the rule (Zhang 2015) 26
Figure 9: Quality control of models with mvdXML filter in xBim, (AEC3, 2017)................ 27
Figure 10: Linked Open Data cloud diagram 2017, (http://lod-cloud.net/) 29
Figure 11: Goal of Linked Data, from Liebich, bSI 2014.. 31
Figure 12: Schema, Instances & Rules, from SIGMOD 2016.. 33
Figure 13: Real Life Object Mapping; Norwegian to bSDD, from bSN2015....................... 35
Figure 14: Setting up a new project, bSN2014 36
Figure 15: configuration and editing within the bSN standard, bSN2014......................... 37
Figure 16: Only the information you need: Filtering and structuring, bSN2014 37
Figure 17: Shape generation in three steps, from Stine and Gips 1972........................... 39
Figure 18: Simplified partial derivation of an existing layout, from Duarte 2005 40
Figure 19: Outcome of physical simulation, from Lorenz et al 2015................................ 40
Figure 20: Numbers of space elements 'n' and their possible solutions, Liggett, 1980..... 41
Figure 21: Project goals on the left, generated designs on the right in Project Akaba 41
Figure 22: Plan layout in project Fractal, buildinglab.. 42
Figure 23: Biomorphs Applet, video screen capture of generation 65 43
Figure 24: Fitness landscape with two genes, Rutten 2010 44
Figure 25: Fitness graph for single parameter (genome), Rutten2011 44
Figure 26: Galapagos Editor, from UMN Digital Design tutorial....................................... 46
Figure 27: First level and second level space boundaries, from Hitchcock et al 2011 47
Figure 28: Proposed Early Design Workflow, from Sleiman et al 2017............................ 50
Figure 29: Top Down (left) versus Bottom Up (right), Streamer D1.6.............................. 51
Figure 30: Properties of a hospital's layers, from NBHI207 55
Figure 31: Perspective Drawing of the Extreme Model (NBHI 2007) 56
Figure 32: Schema of the architect/EDC mvdXML requirement, in the IfcDoc Tool 59
Figure 33: Hourly energy use per day during an average week of an office in a medical centre 63
Figure 34: Sample design rules written in DSL, screen capture from Knowledge editor ... 64
Figure 35: Definition of element rule types in XSD (Hempel & Sleiman) 65
Figure 36: Description of Early Design Configurator, from Streamer review 2017............ 65
Figure 37: Walking distance (left) and Max. separation (right), from Hempel 2016.......... 69
Figure 38: Design validator system architecture, Sleiman2017 70
Figure 39: Internal model of information in EDV (Sleiman 2017 .pptx) 72
Figure 40: Input and output of TECT, from Sleiman et al. 2017 75
Figure 41:Energy simulation results stored for each space... 76
Figure 42: Illustration of KPI Dashboard in the DST, from Streamer D3.6 78
Figure 43: IFC model of the San Luca Vecchio, Careggi... 81
Figure 44: attach labels to rooms with SimpleBim.. 82
Figure 45: property enrichment based on label values with SimpleBim........................... 83

Figure 46: Process flowchart. (Iadanza2009unconventional) 85
Figure 47: Compare function requirement to space property labels................................. 87
Figure 48: Colour by label helps consider room placement (Streamer D1.6) 87
Figure 49: EDC Building Editor for placing constraints (DEMO2017) 89
Figure 50: EDC Design Generator automatically placing rooms (DEMO2017).................. 89
Figure 51: Energy performance in relation to form factor, from Streamer D1.6 92
Figure 52: Increased density hammerhead typology, Cruz & Ortiz, Javakade 94
Figure 53: Youth Detention centre Breda by Hootsmans 96
Figure 54: Stacked playground (l), densely mixed school (r), Oude Dokken school, XDGA 98
Figure 55: Typical floorplan of the Caja Granada Bank, Alberto Campo Baeza 2001....... 101

List of tables

Table 1: Five basic methodology standards, after bSI 2014 17
Table 2: mvdXML tools and their uses 27
Table 3: Namespaces referenced in the ifcOWL 4 ADD2 vocabulary 29
Table 4: Relation between Structured data and data scheme 30
Table 5: Aristotle’s normative sentences (Hjelseth after Sowa, 2000) 32
Table 6: Selected BIM checking tools 34
Table 7: Top-down & bottom-up constraints 53
Table 8: Room label properties and allowed values in the Streamer project 57
Table 9: Streamer labels for different Bouwcollege layers 58
Table 10: Example of rules implemented in the Domain Specific Language (DSL) 63
Table 11: Required input for the EDC 66
Table 12: Extract of EeB solutions table, attachment to Streamer D2.6 73
Table 13: Mapping use classes to Streamer room labels as per room type 83
Table 14: Selected building typologies, after Architects Data, Ernst Neufert 91

Table of Contents

Abstract ... *2*
keywords ... *2*
Italian language abstract ... *3*
Acknowledgements ... *4*
List of acronyms and abbreviations ... *5*
Conventions ... *6*
Definitions ... *7*
List of figures ... *8*
List of tables ... *10*

1 Introduction ... 14

2 Methodology ... 15
2.1 Aims (short term) ... 15
2.2 Goals (long term) ... 15
2.3 Scope of thesis ... 16
2.3.1 Building volumetric aspects ... 16
2.3.2 Formats ... 16

3 State of the Art ... 17
3.1 Interoperability in BIM with open standards ... 17
3.1.1 Terminology, Process, Digital Storage ... 17
3.1.2 The IFC file and its development ... 19
3.1.3 Model View Definitions and LOD ... 23
3.2 Checking; ontology and linked data for semantic reasoning ... 23
3.2.1 Model checking ... 23
3.2.2 LOD & mvdXML, the Completeness check ... 24
3.2.3 Rule checking and the semantic web ... 28
3.2.4 Information Exchange Workflow: the case of Norway ... 35
3.3 Design automation ... 38
3.4 Evolutionary design ... 42
3.5 BIM to Early stage building energy simulation ... 46
3.6 From evidence based design to performance indicators ... 48

4 Semantic BIM design methodology ... 49
4.1 New hospital building in the Streamer workflow ... 49
4.1.1 Why hospitals? ... 49
4.1.2 Brief description of semantic design method. ... 50
4.1.3 Classifying the brief ... 51
4.1.4 Defining design rules from desired relations and from building code ... 61
4.1.5 Configuring Early Designs ... 65
4.1.6 Checking the designs against semantic rules ... 70
4.1.7 Selecting MEP systems & Energy efficient Building solutions ... 73
4.1.8 Comparing design alternatives on Key Performance Indicators (KPI's) ... 74
4.1.9 KPI: Evaluating energy performance ... 75
4.1.10 KPI: Life cycle costing ... 77
4.1.11 Comparing design alternatives with the decision support tool ... 78
4.2 Retrofitting case of San Luca pavilion, Careggi hospital ... 80
4.2.1 Mapping the existing spaces ... 81
4.2.2 Labelling space requirements ... 83
4.2.3 SACS© Careggi database ... 84
4.2.4 Comparing actual space properties vs. function requirements ... 87
4.2.5 Generating design alternatives based on intervention scenarios ... 88
4.2.6 Evaluating energy performance ... 90
4.2.7 Comparing design alternatives with the decision support tool ... 90

4.3 Semantic BIM Design method extended to selected building types ... 91
4.3.1 Collective Housing ... 92
4.3.2 Prisons ... 95
4.3.3 Schools ... 96
4.3.4 Offices ... 99
4.3.5 Chemical Facilities ... 102

5 Discussion and conclusion ... 103
5.1 Summary of main findings ... 103
5.1.1 Philosophy of the method ... 103
5.1.2 Constraints are not subject to optimization ... 104
5.2 Strength and limitations ... 104
5.2.1 Limitations of the human designer ... 104
5.2.2 Retaining explicit knowledge ... 105
5.2.3 Automate repetitive work ... 105
5.2.4 Illusion of the optimum solution ... 105
5.2.5 Design responds to a poorly defined problem ... 106
5.2.6 Design uses intuitive, non-rational methods ... 106
5.3 Conclusions & suggestions for further research ... 107
5.3.1 Iterated interaction between designer and brief ... 107
5.3.2 Reasoning with ifcOWL ontology ... 108

6 Thematic bibliography ... 110
6.1 Building Information Modelling ... 110
6.1.1 Tools: ... 111
6.2 Semantics, Ontologies ... 111
6.2.1 Tools ... 112
6.3 Evidence based design ... 112
6.4 Knowledge based engineering, Design automation, Evolutionary design ... 113
6.4.1 Tools ... 114
6.5 Early stage energy simulation ... 114
6.5.1 Tools: ... 115
6.6 Design Decision Support ... 115
6.6.1 Tools ... 115
6.7 Typlogies ... 115

1 Introduction

Within the engineering field, the building industry stands out for its declining productivity. Knowledge is fragmented between disciplines, leaving knowledge gaps where no-one takes responsibility.

Clients and decision makers expect to have the best possible predictions of building performance. The best choice being the one that most efficiently allocates resources, whilst reducing investment, running costs, while lowering carbon and pollutant emissions.

The early design stage holds most potential for improving design choices, because more choices are still possible. MacLeamy (2004) has argued a Building Information Modelling (BIM) process is suitable for anticipating this effort and reducing cost, see Figure 1.

State of the art developments from different fields hold potential for streamlining the design process, resulting in more efficient buildings, enabled by: knowledge acquisition; computer aided design generation; the availability of interoperable building information models; the use of computer simulations predicting energy consumption; life cycle costing and planning.

The increasing number of requirements that need to be fulfilled by complex buildings can be assisted by information technology. Even more so in the case of buildings with complex space programs, having to satisfy multiple functional connections and very specific space requirements. Examples of such buildings are hospitals, commercial centres, buildings with detention facilities, industrial facilities.

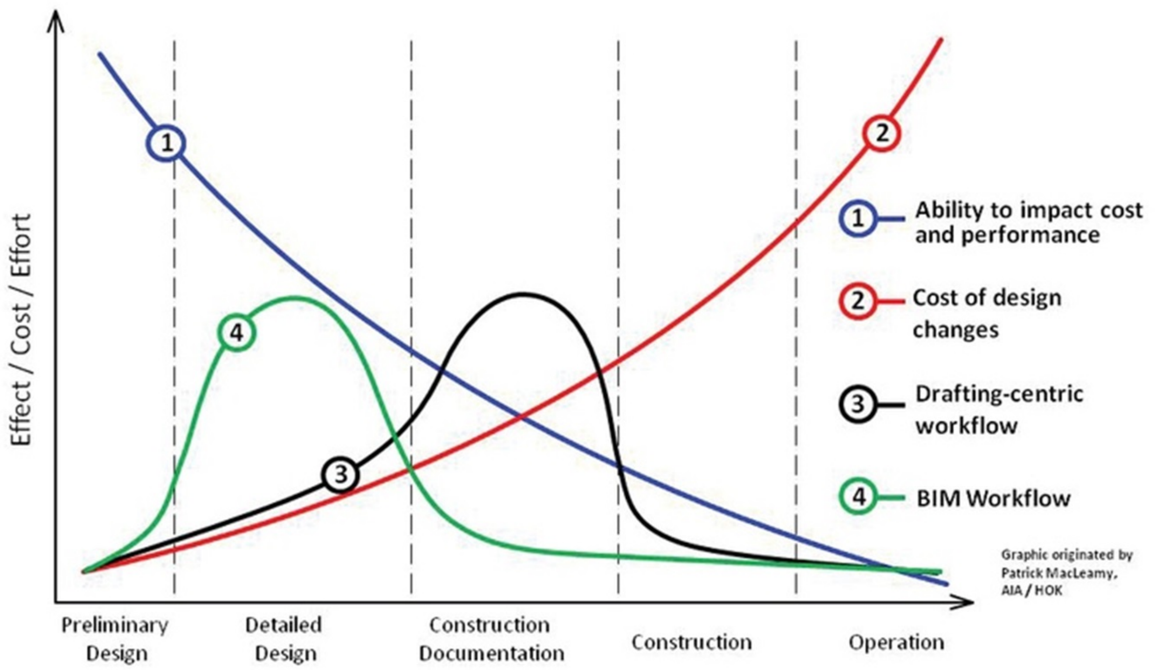

Figure 1: Shifting effect/cost/effort curve with a BIM workflow, MacLeamy (2004)

2 Methodology

This research is structured as follows:

3. Bibliographic research into state of the art technologies that are relevant to the semantic design methodology.
4. Description of researched design methodology:
   i) Description of key steps of Semantic design methodology applied to new buildings in the Streamer project. Designs are compared on their performance in competing areas; cost; energy; quality.
   ii) Alternate method for retrofitting of existing hospital building in the Carreggi hospital case study. The suitability of the existing layout is evaluated, based on the functional requirements for each space.
   iii) Investigation of the semantic design method applied to a range of building typologies: differences are expected in the type of design rules. To understand the limits of the method, I will compare the method to some selected contemporary designs. So as to distinguish between those choices which are likely to be proposed by designers and those which may be proposed automatically.
5. Discussion of philosophy of the method and its strengths and limitations, conclusion and suggestions for further research.
6. Bibliography.

2.1 Aims (short term)

The research aims to describe and connect recent developments across different fields, ranging from semantic knowledge to building information technology, evolutionary design and early stage energy simulation. Speculating how these developments may enable future improvements of the design process.

It also aims to understand the potential impact on the design process of complex buildings. Starting from the need for a well-defined brief, which captures and structures knowledge of domain experts (e.g. client, consultants), it should be accessible to domain experts. It describes a series of instruments that support design rationalization, allowing to maintain an overview of complex boundary conditions to a design. What is the philosophy, what are the strengths and limitations, how can the expert designer interact with the instrument?

Within application of currently available techniques, this thesis seeks to understand the possibilities and limitations as they exist. Secondly, on a more speculative level, it seeks to understand *what part of architectural creativity cannot be foreseen to be replaced by design automation.*

2.2 Goals (long term)

Long-term goals are to improve the quality of the computer assisted design process, and thus improve the designed outcome.

Based on current developments in the fields of semantic web and logic reasoning, the long-term goals in this field of knowledge can be projected as allowing artificial intelligence to reason with building concepts.

2.3 Scope of thesis

The described process method departs from a very clear definition of spatial and functional requirements. This is most feasible in the case of highly standardized typologies, such as hospitals, prisons, courthouses. The design problem in its entirety needs to be well circumscribed. In the process of standardizing requirement and generating layouts, creative possibilities and cultural considerations are -at least temporarily- suspended.
Trying to establish the limits of application is one of the aims of this thesis.

2.3.1 Building volumetric aspects

Outside of the described scope falls the definition of building envelope on an urban volumetric scale. While the proportion of building internal volume to external envelope is one of the determining factors in the energy balance, the investigated methods and instruments are not calibrated on this scale level. The building massing and volume may be predefined for instance by urbanistic zoning rules, or arbitrarily determined on the grounds of energy efficiency or aesthetics by the designer. In the case that this envelope is still free to be defined, it may be shaped by functional considerations, evaluations of solar exposure of public spaces, self shading of building wings, or daylight autonomy of new and existing buildings. Reference can be found in Reinhart for the definition of daylight accessibility guidelines. Recent research by Aicha Diakite looks at the relation between urban masterplans and daylight.

2.3.2 Formats

On a technical level, this thesis will focus on the Industry Foundation Class (IFC) file format supported with mvdXML as the ISO/BuildingSmart standard. While alternative to IFC information exchanges exist through CSV, XML as well as various database connections: ifcOWL; BimQL; bimJSON.

3 State of the Art

3.1 Interoperability in BIM with open standards

Following developments in other industries, the building sector seeks to rationalize workflows to improve information capture and exchange between involved parties, across languages, readable by from humans and machines.

Collaboration in the field of Building Information Modelling is facilitated through the availability of open standards. In the professional field, many experts of Building Information Modelling are in fact software resellers, the larger ones of which tend to understate the importance of shared open standards, claiming instead to offer all solutions in house. The process described in the following thesis document relies heavily on shared processes, formats and definitions to allow the different actors (clients, consultants) to coordinate the information exchange, as such a summary description of the underlying methods seems in place.

The main organization responsible for providing open BIM standards is BuildingSmart International (bSI) for those standards strictly related to building information, the main standards for BIM are built on top of work by the International Standards Organization (ISO), including the EXPRESS-G schema and the STEP physical file format. In the field of sematic data exchange the open web consortium (W3C) is responsible for the Extensible Markup Language (XML), the Resource Description Framework (RDF) and the Web Ontology Language (OWL).

3.1.1 Terminology, Process, Digital Storage

Interoperability in Building Information Modelling is facilitated using Open BIM, defined by BuildingSmart in a series of ISO standards. It was expanded from three parts Process; Model; Terminology, to include change coordination and model view definitions.

Table 1: Five basic methodology standards, after bSI 2014

What it does	Name	Standard
Describes Processes	Information Delivery Manual (IDM)	ISO 29481-1 ISO 29481-2
Transports Information/Data	Industry Foundation Class (IFC)	ISO 16739
Change Coordination	BIM Collaboration Format (BCF)	BuildingSmart BCF
Mapping of Terms	International Framework for Dictionaries (IFD)	ISO 12006-3 bSI Data Dictionaries
Translates processes into technical requirements	Model View Definitions (MVD)	BuildingSmart MVD

Mapping of Terms refers to a standardised uniform meaning, within English language and across other languages. This should guarantee that a building component such as a door, if specified in one language, will be correctly produced in a country with another language.

The BuildingSmart Data Dictionary (BsDD) project aims to collect the concepts, terminology and translations for building information models: http://bsdd.buildingsmart.org/
To facilitate text based exchanges using a variety of protocols, a server with a Representational state transfer (REST) web service is provided at the Uniform Resource Identifier (URI): http://peregrine.catenda.no/

From process analysis to information exchange

The preparation of building project involves many actors that must work together exchanging, reviewing proposals and taking decisions. Typically, actors include a public or private client, consultants, institutions to review permission, (sub)contractors. A first step to rationalising information exchange is the visualisation in a graph of actors, key events and exchanges.
The workflow of building design processes is drawn out graphically using Business Process Modelling Notation (BPMN). An example of a workflow chart is shown in Figure 2.
Defining these processes helps to determine the moments in which specific information sets needs to be exchanged. These are the information exchange requirements, which are then translated in Model View Definitions (see chapter 3.1.3).

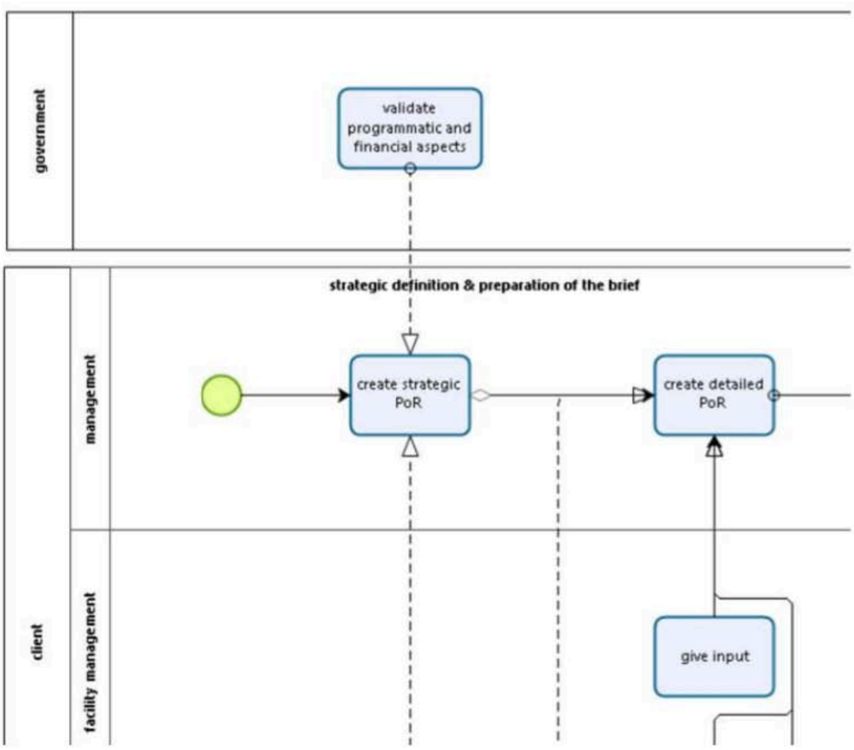

Figure 2: Example: Mapping project initiation using BPMN, from Streamer D5.1

In the future, processes in the building sector are likely to be further streamlined as data analysis methods used in other sectors are applied to the building process (Van Aalst 1998, Van Aalst 2004). According to this method, workflow can be divided in resources, processes and cases. A comprehensive analysis is facilitated by digital information as all events (i.e. emails, appointments, phone calls, etc.) can be recorded. These events can be represented in a graphical mode named Petri nets. It maps routing, which may be sequential, parallel, conditional or iterated. It also registers how steps are "triggered", for example whether it is automatically, by passing of time, by a user or by a message (see Figure 3).

Following the mapping of the process, an expert logical analysis is applied to find and eliminate any inefficiencies found in the existing workflow.

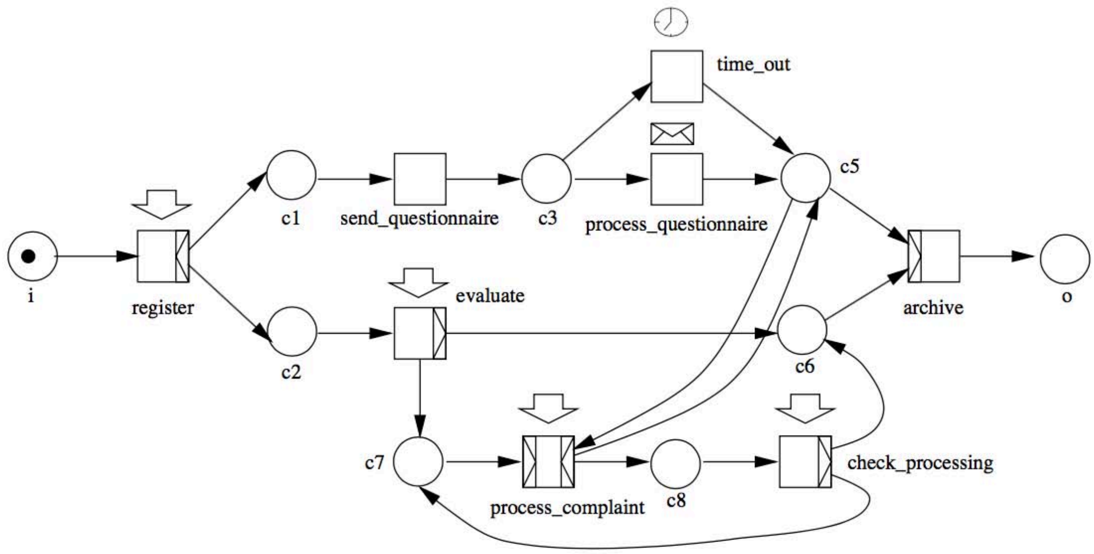

Figure 3: A workflow process extended with triggering information, from Aalst 1998

3.1.2 The IFC file and its development

Many different softwares are capable of producing a Building Information Model within the different disciplines involved: Architecture, Structural Engineering, MEP Engineering, and so forth. File formats for direct data exchange include XML, gbXML, CSV. As a neutral standard, the IFC file allows unbiased control of exchanged information, to which both sender and receiver must comply. Exchanging the geometry or information between different vendors' software's is possible on a one-to-one basis but this would require specific bridges.

The shared standard file format is the IFC file based on the ISO Standard for the Exchange of Product model data or STEP file, defined in *ISO 10303 Industrial automation systems and integration: product data representation and exchange*. Its purpose is the computer-interpretable representation and exchange of product manufacturing information.

An overall framework for information and geometry is defined in the Industry Foundation Classes (IFC). The IFC Schema is a formal specification that can be used by software authors to create the IFC compliant software applications. It is used to represent the structure of information and how that information relates to other information. Meaning is defined in a highly-structured way through concepts, relationships and attributes (as such it is similar to semantics, see chapter 3.2). The data can consist of both geometry and information.
An IFC data set is not necessarily an IFC file, it may also be an XML file, or an access to an IFC server. EXPRESS and STEP are more concise and expressive then ifcXML, but XML is more familiar to most programmers.

In order to understand the IFC data set, a look at the minimum data requirement is helpful. It shall contain a general context reference e.g. project name, author, and only one project. The default units for geometry. The geometric context and geospatial reference are optional. Different aspect models (e.g spatial structure model, architecture, structure, building services domain models) may be represented in different files and contain associations. All should be linked to the same project context.

Within the same model, four types of relations can occur: composition (whole to part relationship); connection (between objects, or between objects and ports, or between objects and features); definition (attaches properties to objects) and association (attaches external references to objects)
Relationships across two or several aspect models include: typing (assigns an object type to one or many object occurrences); containment (provides the spatial containment of building elements or building service elements within the spatial structure); assignment (provides a logical link between objects of different aspect models). This information was summarized from IFC2x4, 4.1.1 fundamental structure of an IFC data set.

Within a project, context units are required and a type library may be defined. A spatial structure is defined (i.e. storey within building on a site), as well as a spatial containment (i.e. wall within a storey). Products may have a placement and one or more shape representations. Objects must have an identification and a revision control and may have an assignment, a composition, connectivity, a grouping, or an object typing. Objects can also be assigned different property sets, which are a collection of properties with pre-defined data types and enumerated values.
Requirements and constraints can be related to control of cost, time, quality and scope. Finally, associations can be made to classifications, documents, libraries, approval and materials. The IFC schema is expandable, in a way that allows user defined additional properties. (from IFC2x4, 4.2 fundamental concepts and assumptions)

http://www.buildingsmart-tech.org/ifc/IFC2x4/rc2/html/requirements.htm#project-library

Data is organised in four schema layers (see Figure 4). The core data schemas, shared element data schemas, domain specific schemas and resource definition data schemas.
It is structured in a modular way, so that underlying concepts can be reused as much as possible.

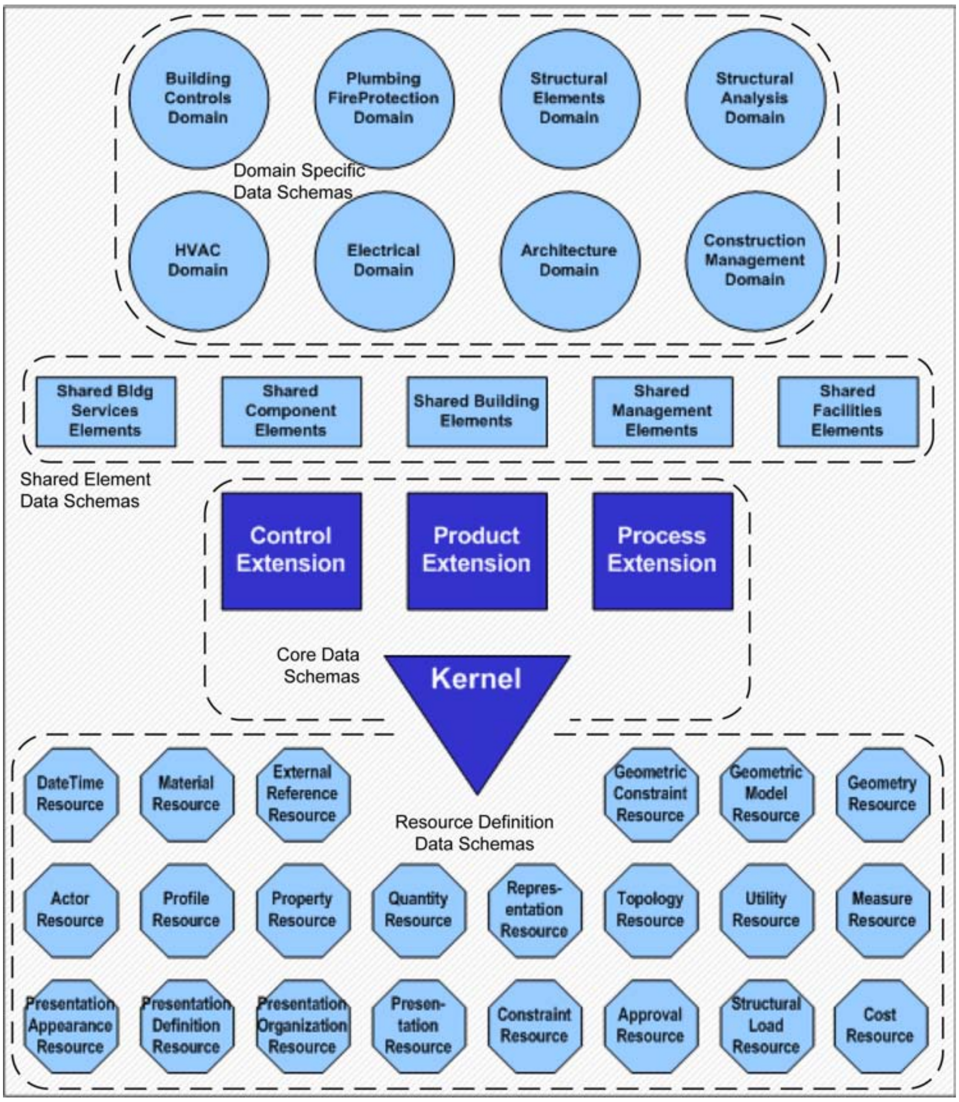

Figure 4: Four layers of data schemas, adapted from bSI IFC2x4 documentation

The EXPRESS language reference manual

In this part a standard product modelling language is defined, EXPRESS, accompanied by the EXPRESS-G graphical notation. The latter is similar to class diagrams in the more commonly used Unified Modelling Language (UML).[1] See the example in Figure 5: Definition of spatial structure elements in EXPRESS-G, IFC2x3_MIG2009.

[1] In fact, the buildingSmart IfcDoc tool now also supports the use of UML.

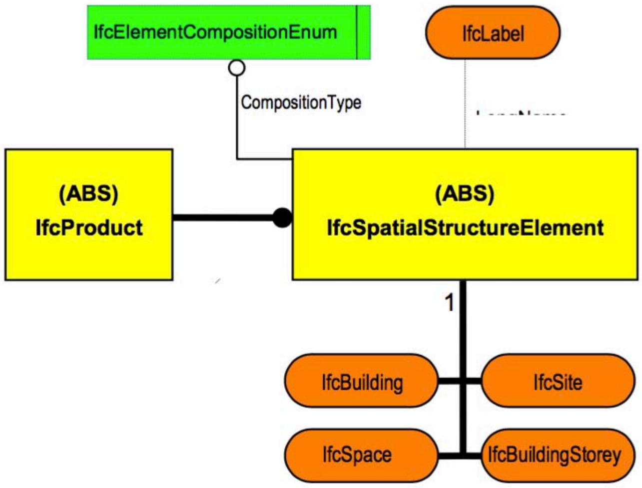

Figure 5: Definition of spatial structure elements in EXPRESS-G, IFC2x3_MIG2009

http://www.buildingsmart-tech.org/ifc/IFC2x4/rc2/html/requirements.htm#spatial-structure

The standards' *implementation methods, clear text encoding of the exchange structure Iconic/ graphic representations*, define a clear text file format, as in the following sample:

```
ENTITY IfcDoor
    SUPERTYPE OF (IfcDoorStandardCase)
    SUBTYPE OF (IfcBuildingElement);
    OverallHeight: OPTIONAL IfcPositiveLengthMeasure;
    OverallWidth : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;
```

In previous versions of the IFC format, when an object occurred in the file more than once the whole object had to be described separately for each occurrence. Now it is possible to refer each occurrence of instances to a single object description. This uses modifiers, such as rotation and local grid references. As a result, the file remains smaller and more manageable. Object entity types may be defined, with their attributes and relationships. Types contain type definitions and can be referenced by several occurrences (otherwise known as instances). Types themselves have no instance. EXPRESS uses object entity types and sub types.

Developments in the IFC4 version include better use of instances, inclusion of electrical appliances, light technical aspects. Future developments will include parametric type objects (Streamer D5.5, 2.1.4), integration for infrastructure and GIS based though a new `IfcAlignment` resource.

3.1.3 Model View Definitions and LOD

Information exchanges often involve partially developed models, in which some buildings components and properties may have been defined more accurately, others less accurately, or may even be completely absent from the model.

The subset of information required in the IFC model for a given information exchange are accurately specified in what is known as a Model View Definition (MVD).

Several early MVD's can be found on the website of the Building Lifecycle Interoperable Software (BLIS) project: http://blis-project.org/.

They have been defined in a shared project by The US General Services Administration (GSA), The Norwegian Directorate of Public Construction and property (Statsbygg) and Finnish real estate agency Senate Properties.

At IFC version 2.0 an initiative named BLIS project, acronym for Building Lifecycle Interoperable Software, has combined the effort of industry experts such as Richard See (Digital Alchemy, Simergy Pro) and Jieri Hietanen (Datacubist SimpleBim), software developers and end user organizations to improve implementations. The approach worked towards implementing the following use case scenarios:

- Design to Design (geometry view)
- Client briefing/space planning to Architectural design
- Architectural design to MEP design
- Arch/MEP Design to Quantities take off / cost estimating
- Arch/MEP Design to Thermal load calculations / MEP system design
- Arch/MEP Design to Construction management/scheduling

On this basis, Model Views and Data Exchange Requirements were defined. Also an Information Delivery Manual was published.

The definition of an MVD is the creation of a partial standard, it requires the input and consensus of many actors and as such it is time consuming to create. The bSI recommendation is to create few agreed MVD's, which can then be referenced when exchanging partial models.

3.2 Checking; ontology and linked data for semantic reasoning

3.2.1 Model checking

When information is exchanged between actors and before proceeding with simulations or such, the model should be checked. Checks can take place at different levels:

1. Syntax;
2. Integrity;
3. Completeness;
4. Semantic rule checking

Syntax check

The first, lower level check is to verify whether the information contained in the file complies with the IFC syntax. A tool specifically designed for this purpose is the IfcCheckingTool, available from the KIT website. It is used also for certification purposes.

3.2.2 LOD & mvdXML, the Completeness check

Level of Detail, Development or Information

Required BIM model contents have been defined in BIM guidelines that do not necessarily refer to bSI/ISO standards. In some countries delivery of proprietary file formats are accepted. Required file contents have been described with Level of Detail, Level of Development (both abbreviated as LOD), combined or alternated with Level of Information (LOI). M. Bolpagni has mapped how these definitions are related and where the differences lie in Figure 5: LoX; global development of LoD, LoI standards.

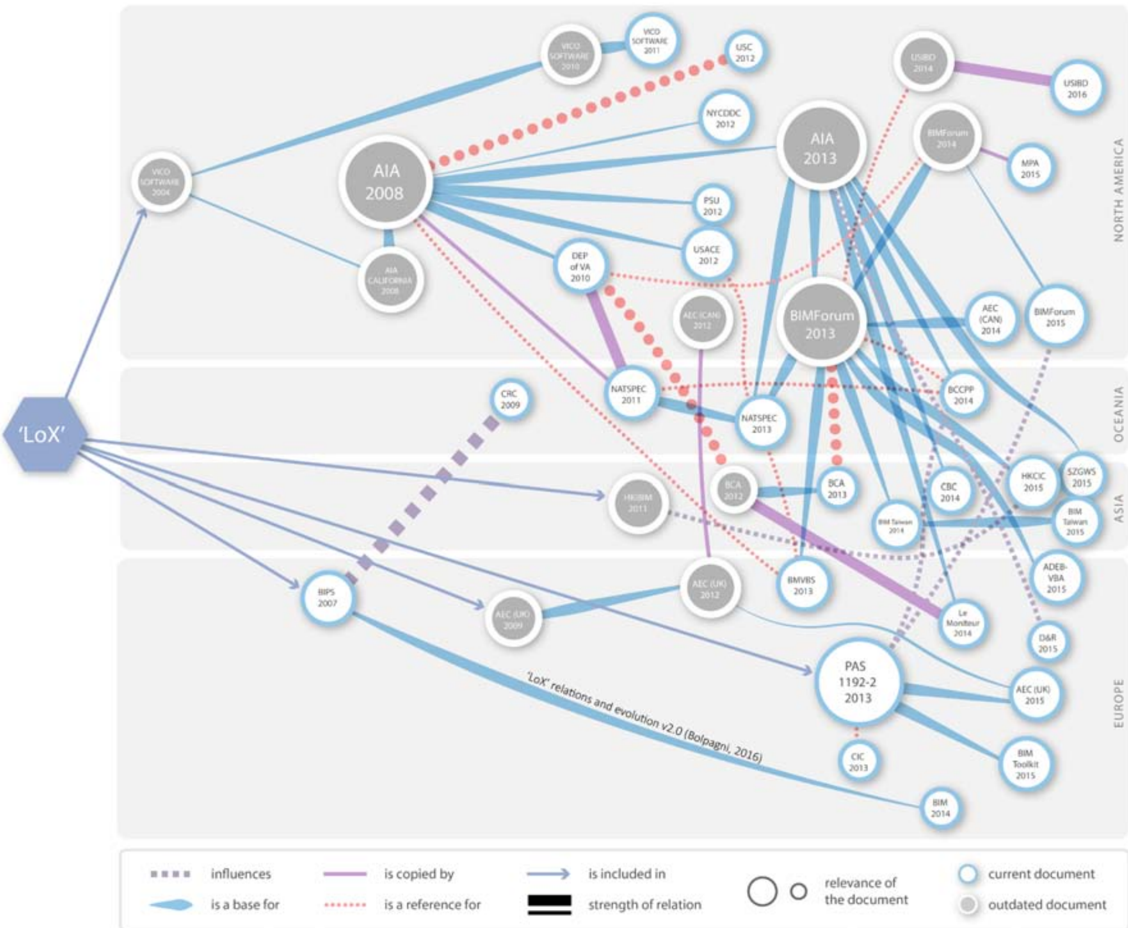

Figure 5: LoX; global development of LoD, LoI standards (M. Bolpagni)

Around the world LOD has been defined in different ways, in Building Information Modeling (e.g. AIA Document E202 ± 2008 Protocol Exhibit) or also in City modelling and CityGML the LODs 0-5.

The LoD definition is difficult to apply as opinions will differ on which LoD the geometry in a model represents, to the point that even the person that created a model is found to be unsure which level LoD it represents. The use as a shared standard is therefore limited (Beetz,…). While the geometry has limits of subjectivity, the exchange of other information is hindered by non-uniform labelling of data. Exchange requirements are often kept in spreadsheet form, often lacking agreement between actors in the process, therefore being time consuming to maintain and not directly suitable for automated use. Consequently, proposals limit the use of LoD to simple requirements for geometry (Treldal), with additionally a focus on information. This additional information may be standardized by country, and by phase (AEC3).

Another approach to Information Exchange Requirements that avoids the drawbacks of the LOD, is the Norwegian Statsbyg BIM Manual (SBM1.21eng). In around a hundred pages, in a tabular format it specifies which information must be present in open BIM files submitted to its national institutions, for example a building code within the municipality, or the cadastral number. It also specifies the data field of the IFC schema shall contain the information, which allows automatic checking whether information is present.

The mvdXML specification

A recent (2011-2013) Buildingsmart specification offers the possibility of translating the Model View Definition into a machine-readable format; the mvdXML.

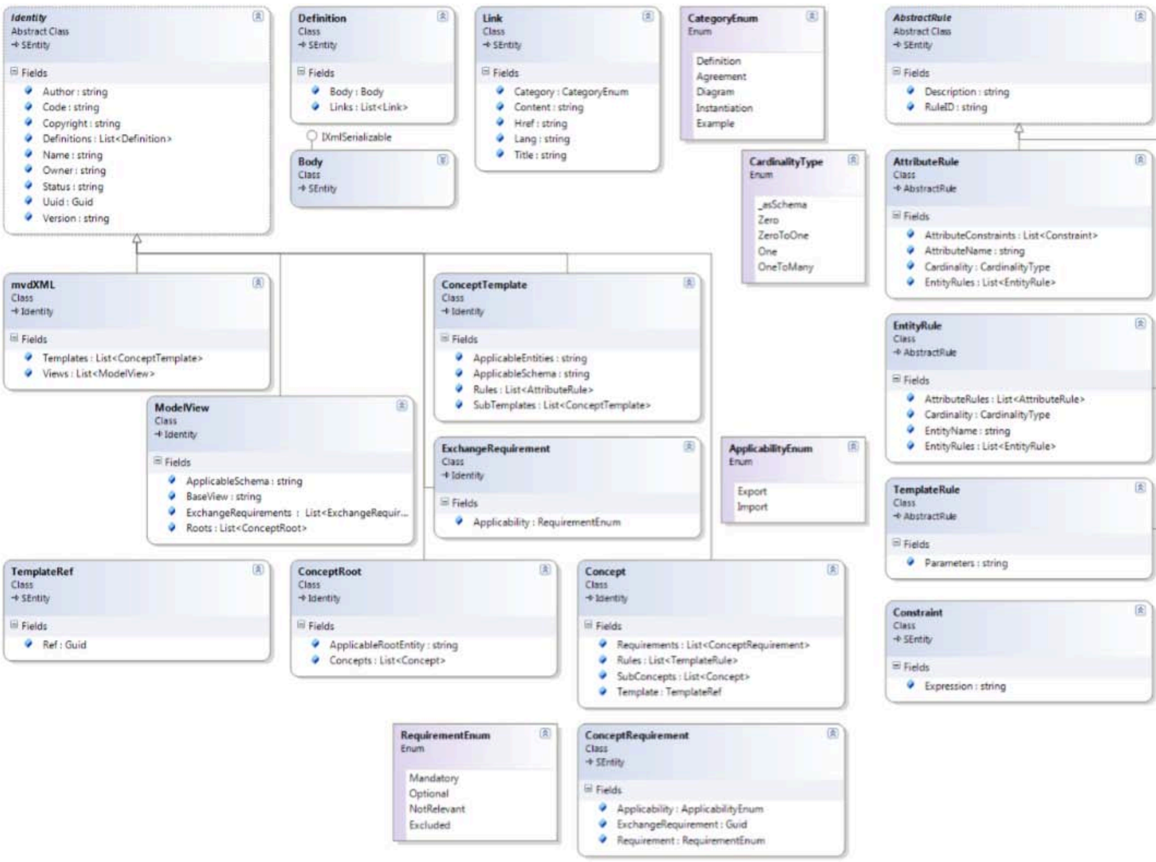

Figure 6: Conceptual model of the mvdXML schema, from bSI MSG 2013

The specification is used to define the MVD subset of the IFC file. Its purpose, elaborated in four use cases, comprises the support of automated IFC model validation, the generation of documentation for specific model views, the filtering of IFC data based on model views, and to limit the scope of IFC to well-defined subsets applicable for applications, such as the exchange requirements used in the Semantic BIM process described in paragraph 4.1.6.
The specification identifies four use cases for the mvdXML standard: documentation, subset schema's, MVD filtering and MVD validation.
The technical specification can be viewed in the referenced document; it is an extensible markup language which refers to a standard XML Schema definition linked by bSI.

mvdXML rules have been defined on the basis of national BIM model requirements laid down in the Rgd BIM Norm and the Statsbygg BIM Manual. The paper gives examples for each kind of rule written both in the notation of logic mathematics as well as a MVD concept template diagram in UML notation. (Zhang2015)

$\forall$ x(IfcElement(x) $\wedge \neg\exists$ y(Decomposes(x, y))

$\supset \exists$ w (ContainedInStructure (x, w)

$\wedge$ IfcRelContainedInSpatialStructure (w)

$\wedge \exists$ z (RelatingStructure(w , z) $\wedge$ IfcBuildingStorey(z))))

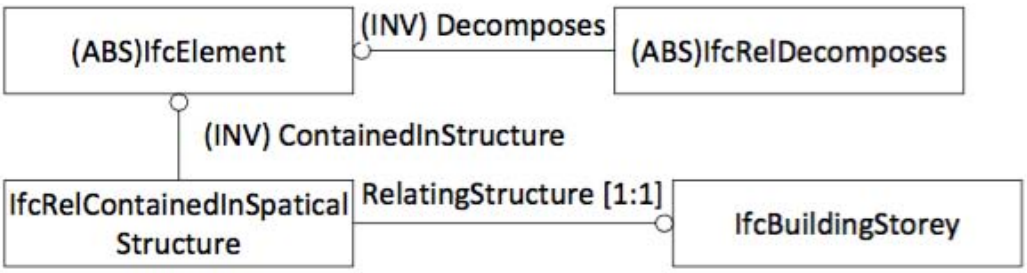

Figure 7: MVD Concept template of the rule (Zhang 2015)

The mvdXML file has a potential use for checking IFC file exchange (bSI Model Support Group 2013, Zhang 2015, Weisse 2016). It can be used to test for data existence and content, however more complicated queries, such as conditional rules rely on the ifcOWL data model, treated in chapter 3.2.

The advantage of testing for pre-defined information requirements is that they can be shared between involved parties and applied automatically. The Model View Definition can be translated into a machine readable mvdXML, for instance by using the freely available IfcDoc tool. The mvdXML checker tool developed by Zhang is available online with a manual. It comes with a set of predefined rules and instructions on modifying those using the IfcDoc tool (Strien 2015). A comparison of mvdXML tools is given in Table 2.

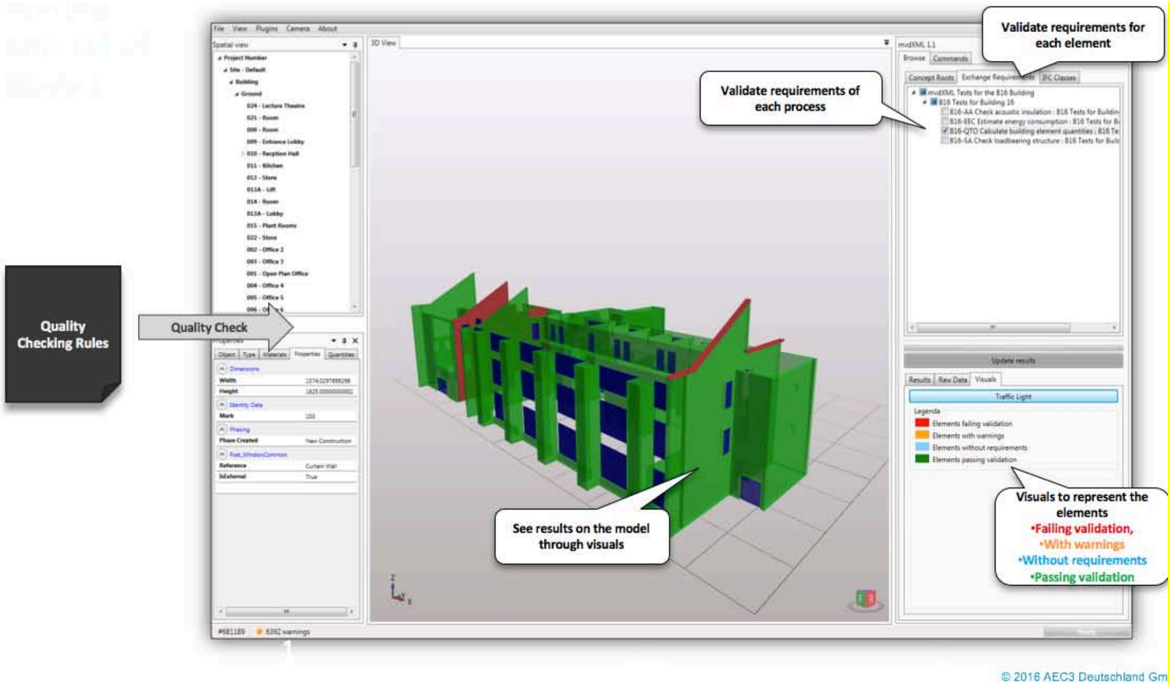

Figure 8: Quality control of models with mvdXML filter in xBim, (AEC3, 2017)

The mvdXML can be composed starting from available MVD's and concept templates in the freely available IfcDoc tool, by buildingSmart.
Bim-Q is an online database for managing exchange requirements.

Table 2: mvdXML tools and their uses

TOOL	DOCUMENT	SUBSET	FILTER	VALIDATE
Bim-Q	●	●		
IfcDoc	●	●		
BIMserver			●	●
Constructivity			●	●
EDM (Jotne EPM)			●	●
eveBIM			●	●
Xbim Xplorer			●	●

More flexible, less structured alternatives have also been developed; to be used in absence of a suitable MVD, Datacubist's SimpleBim can be used to reduce an IFC to only include verified information. In a process described as copy-editing, IFC files may be stripped to include only the information required for the exchange. In this case, unnecessary and unverified information is excluded. Using SimpleBim, the IFC file can also be enriched: for each object with a certain attribute, other properties may be added. While less structured then the mvdXML filter method, the copy-editing permits a lot of flexibility in a process where a formal structure is not agreed. A use case is illustrated in paragraph 4.2.1.

3.2.3 Rule checking and the semantic web

"All knowledge is just a set of statements" (Tim Berners-Lee).

Semantic Web

The most recent developments in BIM elaborate on developments in the field of the Semantic Web, a part of the internet which shares the use of data notation and exchange protocols which allow it to be machine readable. These standards are maintained by the open web consortium (W3C) and referred to as the Semantic Web. While conventional html web pages might only identify graphical or layout information, content on the semantic web has meaning assigned and relationships. This content is referred to as Linked Data. The linked data contains references to standard definitions that are shared online.

As it is defined by its relations, the Berners-Lee has also referred to the semantic web as a global graph. Linked open data (LOD) is what makes data available and searchable, what gives context.

The inventor of the internet rates linked open data according to the following rating (Berners-Lee2009):

★	*Available on the web (whatever format) but with an open licence, to be Open Data*
★★	*Available as machine-readable structured data (e.g. excel instead of image scan of a table)*
★★★	*as (2) plus non-proprietary format (e.g. CSV instead of excel)*
★★★★	*All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff*
★★★★★	*All the above, plus: Link your data to other people's data to provide context*

The standard framework for defining data on the Semantic Web is the Resource Description Framework (RDF), first published by the W3C consortium. Originally it was only intended for metadata. It has since been generalized to work for web based, graph databases.

RDF is often represented in forms of triples (Subject, Predicate and Object). We may also think of these triples as Attribute, Relationship, Object, rather than Subject, Predicate, Object, the predicate can be thought of as a property or even a verb, as in the following example of N3 notation:

```
<#patrick> <#knows> <#joe>
```

N3 is a human readable format for serializing OWL, but it's just one of several formats available for serializing OWL, while others are OWL/XML, RDF/XML, Turtle. These *Triple Stores* can be accessed with Query Languages, or submitted to semantic reasoners.

Where possible it is preferred to refer to classes of objects or concepts that have been defined elsewhere. This increases the interoperability of any given data. The use of a International Resource Identifier (IRI), a type of Uniform Resource Identifier (URI) permits linking to definitions that are shared online, for example for bibliographic metadata. Referenced vocabularies are listed in table Table 3. The use of a namespace as a prefix reduces the lengthç in the following example the definition of the concept *title* is found at the given URL, marked by the *dc* prefix:

```
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
<> dc:title
  "Primer - Getting into the Semantic Web and RDF using N3"
```

Table 3: Namespaces referenced in the ifcOWL 4 ADD2 vocabulary

Namespace	IRI/URI
cc	<http://creativecommons.org/ns#>
owl	<http://www.w3.org/2002/07/owl#>
rdfs	<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
list	<https://w3id.org/list#>
xsd	<http://www.w3.org/2001/XMLSchema#>
expr	<https://w3id.org/express#>
rdfs	<http://www.w3.org/2000/01/rdf-schema#>
vann	<http://purl.org/vocab/vann/>
ifc	<http://www.buildingsmart-tech.org/ifcOWL/IFC4_ADD2#>
dc	<http://purl.org/dc/elements/1.1/>

While the IFC format describes the entire domain of building structure and process, much information relevant to the process may be available outside of this domain. Adapting technologies developed for the Semantic Web, such as ontology, querying and linked data standards, allows creating connections between BIM data and available data outside of the BIM domain, such as: Geo Information Systems (GIS), building materials, building products, live sensors, rules and constraints (Liebich, bSI Toronto 2014, Pauwels bSI Barcelona 2017). A sample of 1139, as of January 2017, is shown in Figure 10. Within this graph, the Cross Domain category includes databases linking to other databases, such as Freebase and DBpedia.

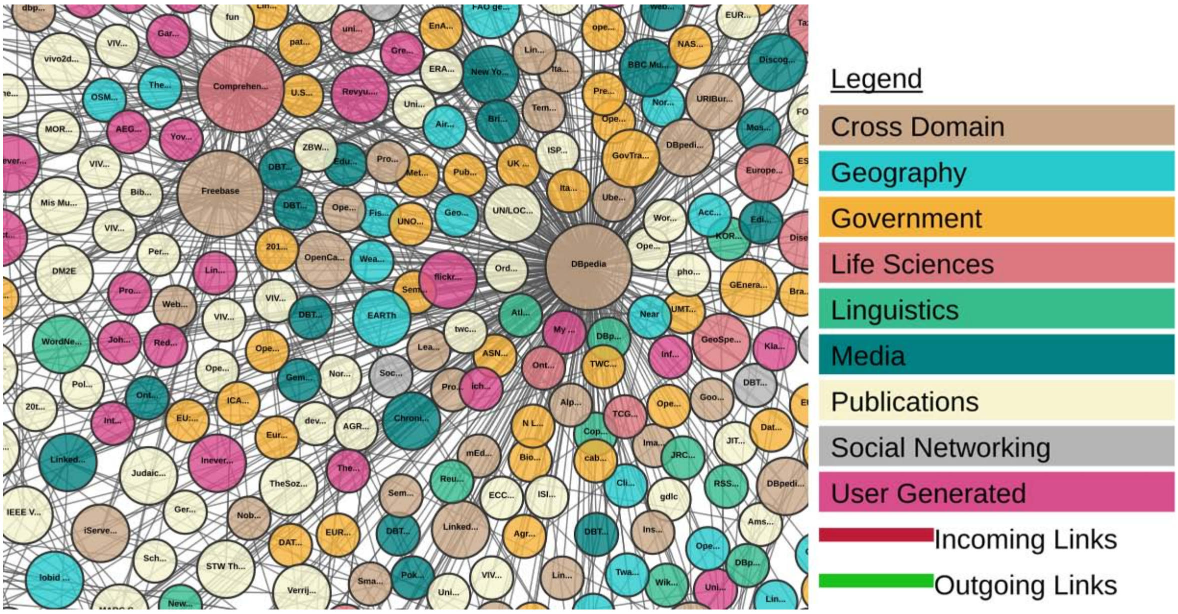

Figure 9: Linked Open Data cloud diagram 2017, (http://lod-cloud.net/)

The ifcOWL ontology

The new standard Web Ontology Language for IFC (ifcOWL), developed by the bSI Linked Data Workgroup, connects the building domain to the Semantic Web. This enables two new actions, access to linked data in available databases outside of the building domain that contain relevant knowledge

Logics and reasoning require that the meaning of concepts has been explicitly defined; this is referred to as an ontology. In the introduction of his 2012 book Maedche explains the meaning and purpose of ontology: Ontologies are methods of structuring by assigning explicit meaning. Structuring the knowledge of "domain" experts allows this knowledge to be reused. The domain in this case could refer to the Architecture, MEP, or Construction domain. In order to prepare machine readable data exchange, *"ontologies capture the structure of a domain"* (Streamer D5.1, p13)

"Conceptual structures that define an underlying ontology are germane to the idea of machine processable data on the Semantic Web. Ontologies are (meta)data schemas, providing a controlled vocabulary of concepts, each with an explicitly defined and machine processable semantics**. By defining shared and common domain theories, ontologies help both people and machines to communicate concisely**, supporting the exchange of semantics and not only syntax. …." (MaedcheStaab2012)

"Ontologies are a crucial technique to formally describe the objects of a certain knowledge domain. An ontology normally defines a number of concepts and a set of rules, logically relating concepts to an ontological network. By means of formal logic, this system supports automatic reasoning." (Haefele2015)

Ultimately ontologies are a crucial step preparing for knowledge acquisition form a preparation for processing with artificial intelligence (Maedche2012).

IfcOWL, Model and tools for the conversion of the IFC based on the EXPRESS schema to an RDF graphs based on the IfcOWL schema (Terkaj W, Sojic A 2015), Pieter Pauwels and Walter Terkaj 2016). The standard of ifcOWL has been presented for approval at the 2015 bSI summit in Singapore.

Table 4 shows the equivalents for structured data and data scheme (after Pauwels).

Table 4: Relation between Structured data and data scheme

Structured data	Data scheme
IFC Step Physical File (IFC SPF)	EXPRESS
Extensible Markup Language (XML)	XML Schema Definition (XSD)
Resource Description Framework (RDF)	IFC Web Ontology Language (ifcOWL)

Several tools to convert from IFC to ifcOWL are offered by the authors of the ifcOWL standard Pauwels and Terkaj (bSI, Barcelona 2017). The authors claim a fully successful roundtrip, which means identical files before and after conversion. However, the many possibilities of geometry representations in IFC still need to be translated into OWL, a proposal was made by Pauwels, Krijnen, Terkaj and Beetz (2017).

Each of the IFC terms are mapped to OWL concepts and constraints, each is linked to an Internationalized Resource Identifier (IRI), for example:

http://www.buildingsmart-tech.org/ifcOWL/IFC4_ADD2#IfcSpatialStructureElement

The ifcOWL vocabulary now allows linking data to information outside of the BIM domain, as shown in Figure 10: Goal of Linked Data, from Liebich, bSI 2014.

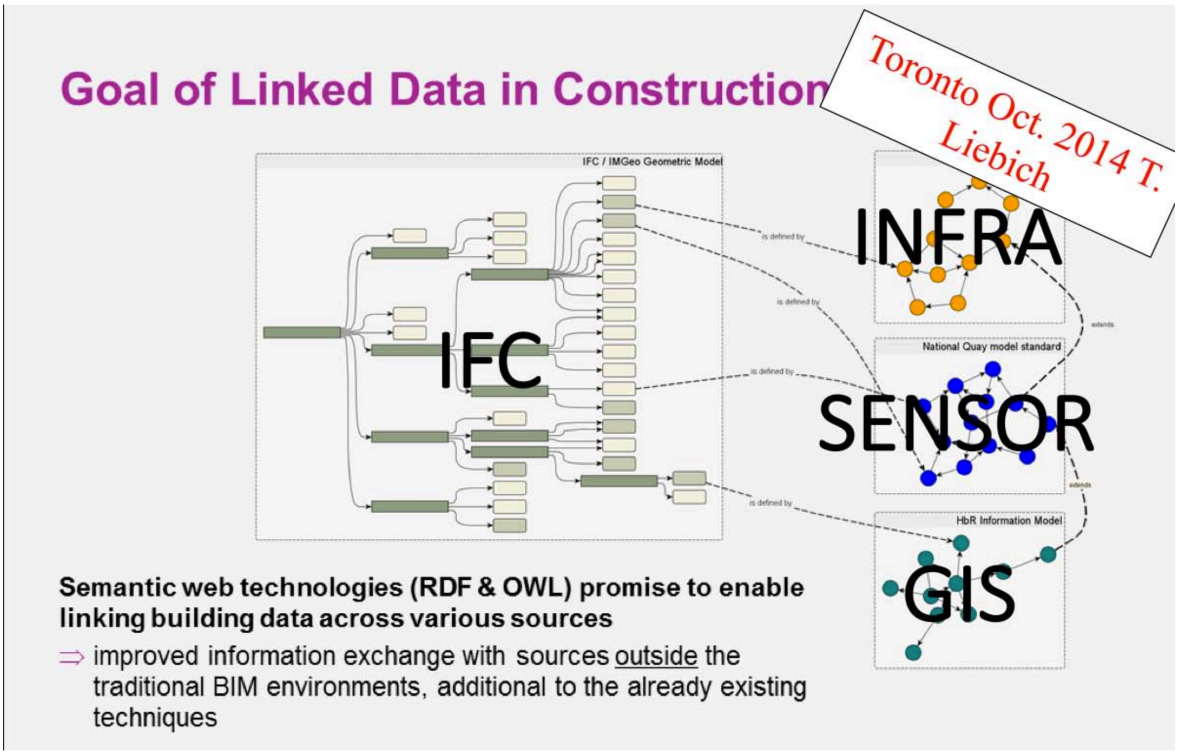

Figure 10: Goal of Linked Data, from Liebich, bSI 2014

Example use cases presented by Pauwels at BSI BCN2017: Linked data for infrastructure: expanded ontology for bridge parts, infra objects, measures, sensor data from monitors, geo context, rules and constraints;
Simplified Model Views from simplified building graphs, uses a query to obtain a model subset. ifcOWL queries can also be used to obtain subsets (or Model Views) of an IFC model as demonstrated by Roxin and Weise.

Semantic rule checking & Logic reasoning

The checking methods treated up to this point all contain limitations, as explained by Zhang and Beetz. *"The evolving mvdXML specifcation is currently the only open standard used to formally capture MVDs and validating IFC instances (Chipman et al, 2013). Its built-in constructs are used to develop reusable subsets and constraints. However, these modelling constructs specify constraints on low-level IFC entities and attributes in its underlying schema, which make them more suitable for BIM certification and are not easily reusable by the general public." (Zhang2015)*

An established mode of BIM model verification uses Solibri Model Checker, which can verify the building information and geometry contained in the IFC against logical rulesets. These rulesets may be predefined or can also be adapted by the user. As a proprietary check the rules have the limitation of not being available to all stakeholders involved in a project.
"… Although [Solibri] offers partial model extraction, sophisticated queries and constraint checks, these mechanisms are not based on open, reusable specifications and cannot be tailored to individual needs in straight-forward, non-proprietary ways." (Mazairac2012)

"A number of these examples rely on semantic data enrichment or schema and data transformations. Namely, the in- formation needs to be used in a syntax and semantics that is different from the one provided (typically IFC)."
In response to these limitations several methods have been developed. Developments in research have attempted to expand the use from proprietary rules to openly available rules. Following established mechanisms in Semantic Reasoning, which use Inference to add knowledge.

The field of logic mathematics and reasoning is mature so that certain types of deductions can be proven to be true, at the simplest level this works like the following "truth" table, Table 5.

Table 5: Aristotle's normative sentences (Hjelseth after Sowa, 2000)

	Affirmation	**Denial**
Universal	Every A is B	No A is B
Particular	Some A is B	Not every A is B

The highest level of checking uses semantic rules to verify logical relations between entities in the model. While the mvdXML is applied to all defined objects, semantic reasoning can analyse and elaborate on single objects and their mutual relations.
Furthermore, it will allow the use of Semantic reasoners, which are computer software capable of inferring logical consequences from a set of asserted facts or axioms. This is the domain of logic or logic programming. Many use first predicate logic, based on inference, uses ontology and logic rules to enrich available information.

For example, if an MVD can require that a window needs to contain the attribute `isExternal` (with a value that may be Yes or No), Semantic reasoning can infer a window `isExternal: Yes,` if it is placed in a wall that has the attribute `isExternal,` or that has External in its name. A future step for this example, for this example, would be to deduce the `isExternal` attributes' value from the topological placement of the window relatively to other elements e.g. ifcSpaces, in the IFC file.

Research by Zhang and Beetz (2015) has translated eighty percent of Statsbygg BIM Manual into semantic rules.

Attempts are to use modularity, wherein defined concepts can be re-used, the consistent use of concepts can be guaranteed by:

1. the use of the bSDD in which all concepts are uniquely numbered and specifically declared;
2. the ontology standard which maps the IFC EXPRESS schema to ifcOWL

An example use case is the formulation of queries derived from regulations: e.g. select all stairs that qualify as an emergency exit (Roxin2016).

Finally, several authors have compared the semantic reasoners SPIN VM, EYE and Stardog to establish a baseline reference for query time performance. For a large scale query performance benchmark, a total of 369 publicly available BIM models have been queried. (SIGMOD, Semantic Big Data Workshop, 2016).

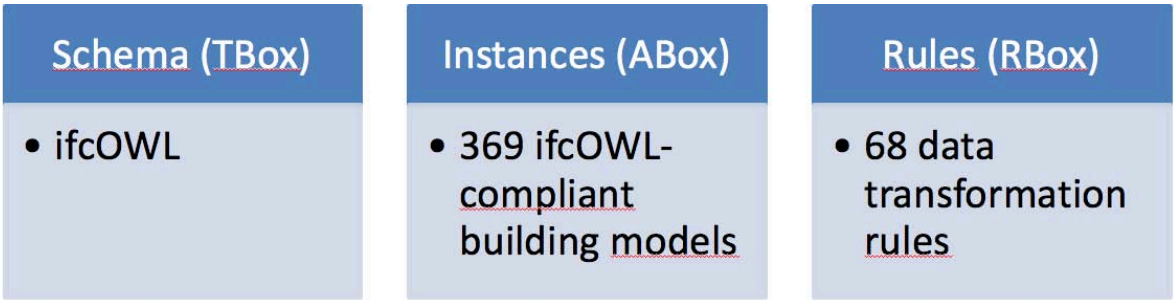

Figure 11: Schema, Instances & Rules, from SIGMOD 2016

"At the core of the (…) approaches are three key components:

- *(the TBox), a schema (OWL ontology) that defines what kind of information is used by the rule checking process and how it is structured;*
- *(the ABox) a set of instances (RDF graphs) asserting facts based on the concepts defined in the TBox and;*
- *(the RBox) a set of rules (e.g. IF-THEN statements) that can be directly combined with the schema).*

"Declarative data transformation procedures are then accessible as soon as all the data and all the rules are available in a complete and consistent shape: inferences are generated by generic reasoning engines, the results are asserted as new facts into the graph, after which they are used in specific applications (e.g. simple visualisation in a graphical user interface; job hazard analysis; acoustic building performance checking). Depending on the rules that are being triggered, one set of information then has the potential to be made available in a diverse number of forms (…), bringing an entirely new form of interoperability for an industry that has always relied heavily on the combination of an agreed standard with many in transparent import and export procedures that were implemented using procedural programming languages. Namely, with its logical basis in Description Logics (DL), a logic-based semantic big data publishing approach emerges." (Pauwels et al SIGMOD2016)

Tools

EXPRESS-to-OWL and IFC-to-RDF conversions can be made with the following tools:

IFCtoRDF, Implementation in open source JAVA project.

ExpressToOwl closed source C++ tool to automatically convert an EXPRESS schema into an OWL ontology.

IfcDoc Implementation in open source .NET project.

BimServer EXPRESS-to-OWL and IFC-to-RDF conversions can be made with a plug in. The server also has BimQL implemented: A BIM specific query language defined to meet the following requirements: the possibility to define custom queries; user friendly; platform independent and open source. (Mazairac 2013, Zhang et al 2015).

Stardog is a graph knowledge editor which accepts user defined rules and imports OWL schemes.

Topbraid Editor, Server and API use SPARQL Inferencing Notation (SPIN) which build on the W3C SPARQL standard. *"These vocabularies let you define new functions, stored procedures, constraint checking, and inferencing rules for your Semantic Web models".*

ModelcheckN3 is an experimental semantic software tool developed specifically for BIM, it is a low threshold open source query editor using an Euler reasoner and N3/Turtle notation (Helm and Berlo 2015). Developed prior to the finalizing of the ifcOWL standard.

Table 6: Selected BIM checking tools

TOOL	SYNTAX CHECK	INTEGRITY CHECK	MVDXML CHECK	SEMANTIC RULE CHK
BIMSERVER	●	●	●	●
CONSTRUCTIVITY VIEWER	●	●		
IFCCHECKINGTOOL	●			
MVDXML CHECKER				
SOLIBRI MODEL CHECKER	●			●
XBIM WITH MVDXML PLUGIN	●	●	●	
TOPBRAID*				●
STARDOG*				●
MODELCHECKN3*				●
*)REQUIRE ifcOWL				

3.2.4 Information Exchange Workflow: the case of Norway

BuildingSmart Norge (Norway) released in January 2015 a standard for managing the information exchange in BIM processes: https://buildingsmart.no
It uses the Bim-Q platform (a product by AEC3), which has been adopted as part of a national standard (bSN2014-bSN2015). Bim-Q is a database to collect exchange requirements from Domain experts.

Structured lists of exchange requirements and delivery of open BIM files were already obligatory under the Norwegian Statsbyg BIM Manual (SBM1.21eng, see page 54) by the Norwegian Directorate of Public Construction and Property. As these lists follow the IFC data schema, they are ready for translation into an ifcXML data requirement.

The next step towards developing a national system was for BuildingSmart Norge to create a nationally agreed translation of used terms in the building process in a matrix called Real Life Object Mapping (RLOM); for compiling the brief, obtaining permits and constructing, also linking these terms to the international standard (see also bSDD, page 17).

NS 3451 Code	bSDD guid	Real life object name	RLO description Norwegian	IFC4 entity	IFC4 type	Ifc4 Enumeration
214	1RFYC0WJeHu000025QrE$V	**avstivende vegg**	Omfatter andre avstivninger både permanente og midlertidige	IfcWall	IfcWallTypeEnum	SHEAR
214	0e9pE0BS4Htm00025QrE$V	**stålspunt stålspuntnål**	Spuntvegger både permanente og midlertidige.	IfcPile	IfcPileConstructionEnum	PREFAB_STEEL
214	0fMS7pOa11tgGGlp9khkl9	**støttemur**	Omfatter andre avstivninger både permanente og midlertidige).	IfcWall		

Figure 12: Real Life Object Mapping; Norwegian to bSDD, from bSN2015

"RLOM establish a standard relation between standardized object types in Norwegian specifier language and IFC. 800+ object types identified, so far..." (November 2015)
This way professionals can work with object types in own language and IFC exports can be standardized on National level. Note that the second column of the table collects the bSDD unique identifier (GUID) for each term or concept, originally defined in English. The third column links this to the corresponding Norwegian term.

BuildingSmart Norge Guiden consists of a national database, provided with templates of MVD's for different pre-defined exchange requirements. It is customizable for each process: The mvdXML filters can be tailored to each project's need. It is possible to select pre-defined Phases; Procedures; Rolls (Figure 13: Setting up a new project, bSN2014).
The national database brings together client needs, domain expert knowledge (Client, Architect, Engineer,…) and modelling expert knowledge. Exchange requirements are shared online in a database built on reusable templates, agreed nationwide. The aim is to reuse the rules by creating them from templates. Upon compiling the process data an mvdXML, filters are made available for download for each phase, which can then be used by all involved for an automated completeness check.

The tasks for each roll have been defined, and a teaching and certification programme guarantees the correct application.

As far as the author is aware, this system is unique, for its completeness, modularity and BIM standard based workflow. Unfortunately, the functioning of the entire system has not been translated into English, German or French. Therefore, it is not getting as much attention as it would deserve, or for instance as the much-admired British BIM roadmap is getting.

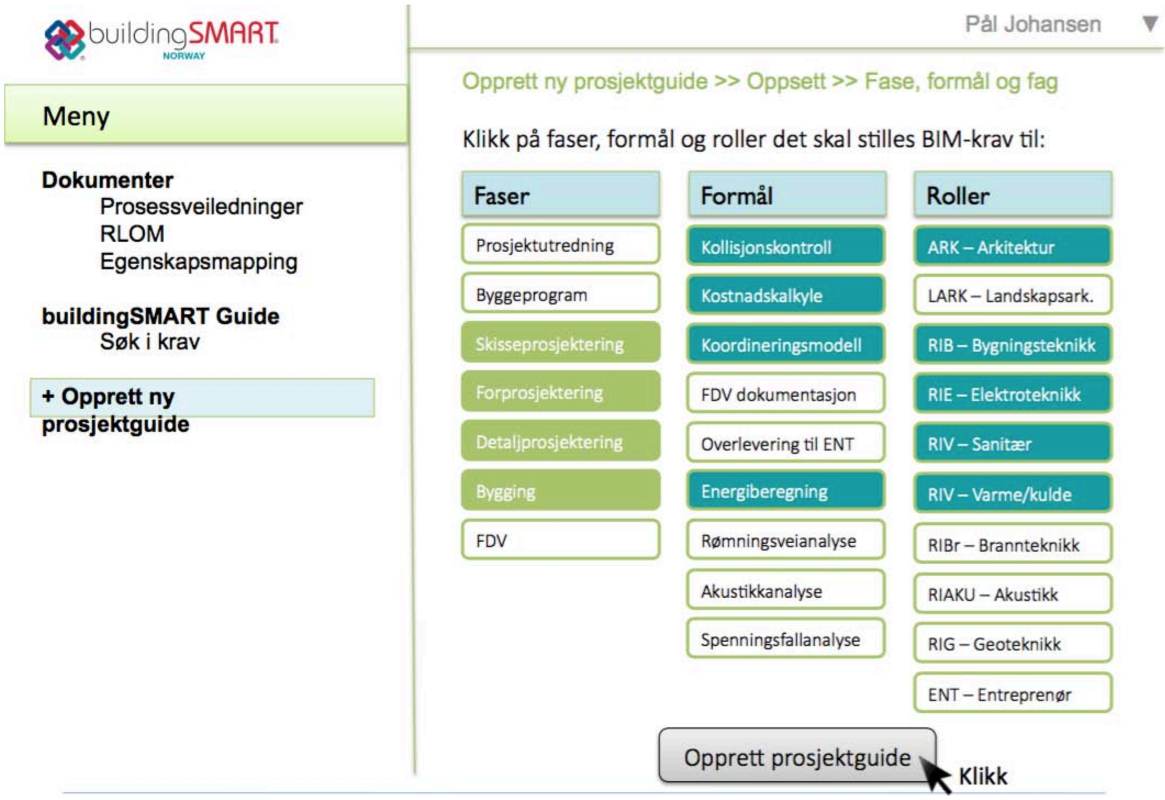

Figure 13: Setting up a new project, bSN2014

Screenshot of the bSN Guide portal, showing how a user can tailor the information requirements to their project's need: The user can select Phases; Procedures; Rolls.

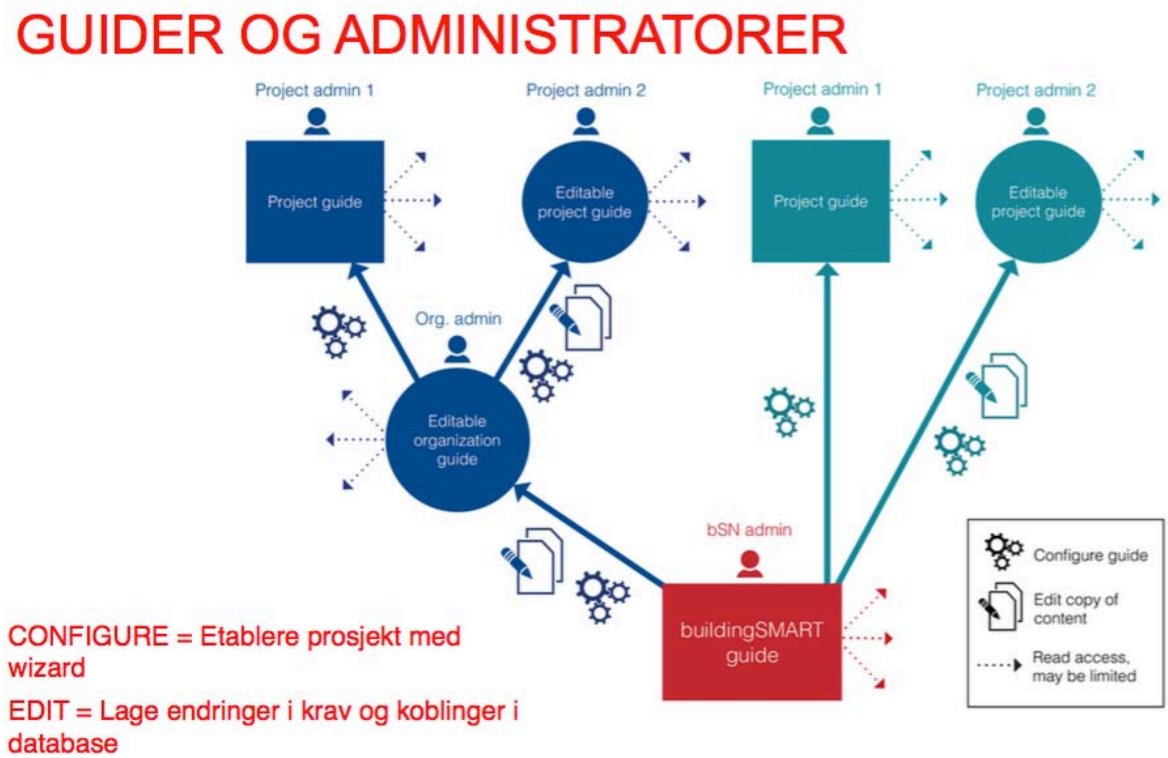

Figure 14: configuration and editing within the bSN standard, bSN2014

Configure: Establish project with wizard. Edit: Make changes to requirements and links in database

Figure 15: Only the information you need: Filtering and structuring, bSN2014

Image to represent how the standard was designed to not confront users with technical information they do not strictly need.

3.3 Design automation

Beyond the semantic rules described in the previous chapter, which are limited to the purposes of verification of a given design, rules may also be used as input for generating design proposals. In their book, *Design Rules: The power of modularity* (2000) Baldwin and Clark look at design from an engineering point of view. They explain how design parameters can be conflictual and working in opposite directions. Designing within this context sometimes requires an iterative approach. Traditional design processes might use undocumented "Rules of Thumb" which may lose relevance in new conditions.

They propose modularity as a way to deal with complexity, with the following objectives:

- *To make complexity manageable;*
- *To enable parallel work; and*
- *To accommodate future uncertainty.*

In the process, they describe products should be designed as a composition of modules; to them design rules are the rules that guarantee that independently designed modules fit together. In the car industry and in electronics this approach is demonstrated to be efficient and flexible.
According to LaRocca 2011, Knowledge based Engineering (KBE) is a technology based on dedicated software tools called KBE systems that are able to capture and reuse product and process engineering knowledge. The main objective of KBE is the reduction of time and costs, and to improve the quality of product development by means of the following:

- Automation of repetitive, non-creative, design tasks,
- Support of multidisciplinary design optimization in all the phases of the design process"

Currently in BIM software, some degree of automation is achievable by a combination of scripts or visual programming in combination with building components with physical properties (resistance for electrical parts, or pressure loss for plumbing parts).

A more extensive interpretation of design rules is as a set of drivers of design choices in an automated process called Rule Based Design. The design rules need to capture the tacit knowledge of domain experts (In this case the Client, Architectural, Engineering & Construction domains). To use the rules as input for a programme they need to be calculable, i.e. based on numeric formulas or logic relations.
Advantages include that regulations can be processed as rules, that the design input remains transparent throughout the process open to re-evaluation (for instance if a change in design conditions require a new design iteration), the designers' personal knowledge and preferences can be recognized as input. As each individual designer makes choices between design alternatives based on cultural and personal preference.

Automated design has been developed for repetitive engineering tasks with clear boundary conditions. In fact, early fields of application include the routing of industrial piping and the design of electric circuits. The design questions in the field of architecture have not been defined as clearly. One relatively clear set of requirements can be defined for the adjacency and access relations between rooms, at least for some building types. That is why the design of floorplan layouts has seen many attempts at automation.

In the field of architecture there are several different approaches to floorplan design (Schneider, Koenig et al 2010). Each approach has drawbacks

- Constraint-Based Systems
- Cellular automata and agent-based systems
- Shape Grammars
- Physically-Based Systems
- Evolutionary Algorithms

Constraint Based Systems start from clearly defined relation requirements between rooms. This may be in tabular format, or graphically as a functional relation scheme. These approaches use the topology from the relation scheme as a starting point for generating floor plans.
Cellular automata and agent based systems simulate processes and are bound to very specific geometries. They are thus more relevant to urban planning than to architectural floorplans.

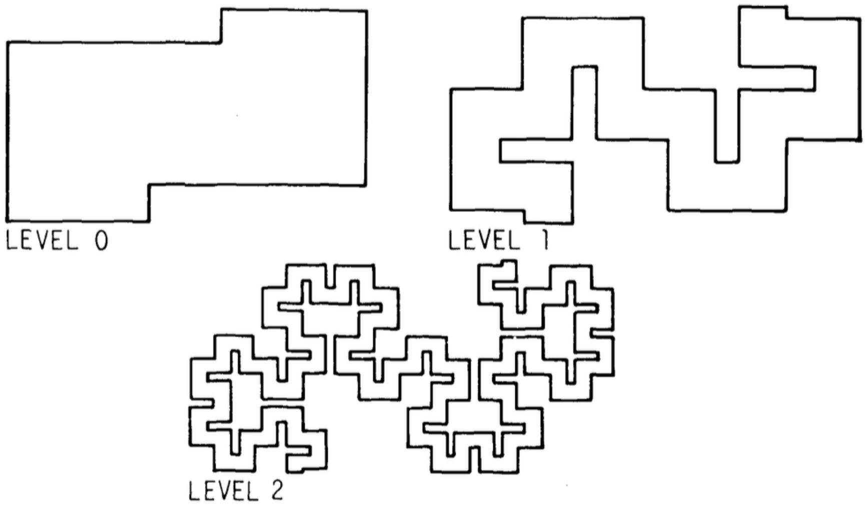

Figure 16: Shape generation in three steps, from Stine and Gips 1972

Then there exists the shape grammar approach, which uses compositional algorithms to explore design options. First presented as a graphical method generating shapes similar to fractals by repeating the same formal operations on shapes on different scales (see Figure 16: Shape generation in three steps, from Stine and Gips 1972).
Shape grammars have been developed by others since, for instance by Duarte to recreate floorplans of housing by Alvaro Siza, see Figure 18.
The shape grammar method works by subdividing spaces, in consecutive steps following many compositional rules, regarding orientation, articulation, spaciousness, topology, proportion, etc. Each floorplan comes with a topological scheme. Duarte’s method will not calculate the best option, but will generate all possible configurations within a given set of rules, followed by a process of eliminating the worst options one-by-one until the best option remains (Duarte2005).

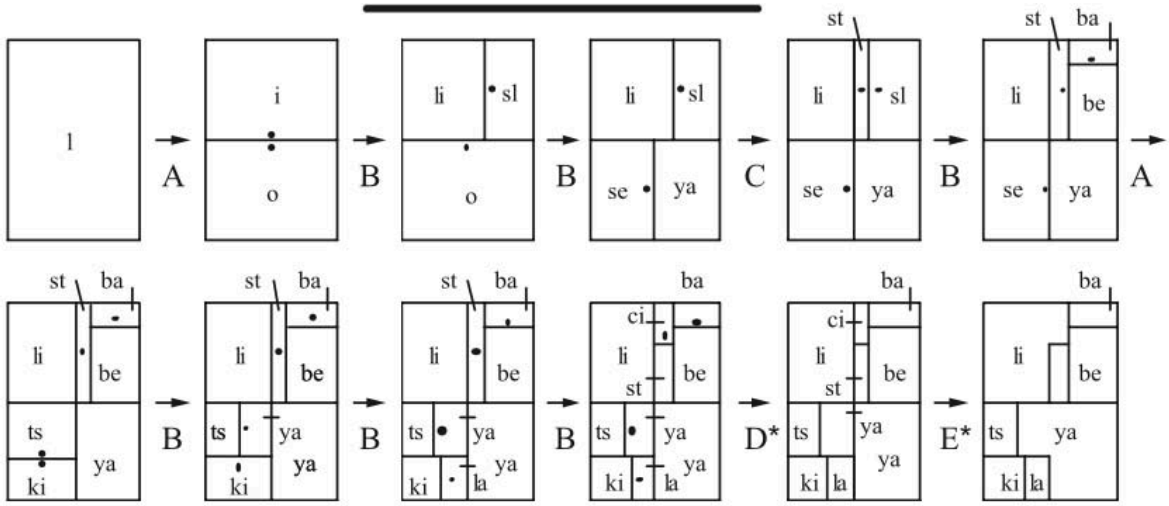

Figure 17: Simplified partial derivation of an existing layout, from Duarte 2005

"There are rules for dissection (A, B, and C), connecting (E), and extending (F) rectangles. The remaining rules are for deleting a marker (D), assigning a function (G), and permuting functions (H). Legend: l—lot, i—inside zone, o—outside zone, li—living zone, sl—sleeping zone, se—service zone, ya—yard zone, be— bedroom, ba—bathroom, ki—kitchen, ts—transitional space, la—laundry, pa—pantry, ci—circulation, st—stairs. The asterisk means that the same rule was applied several times."

Physically-Based Systems are based on physical models, attraction can be modelled by connecting room objects with virtual springs that pull them closer together. The edges of rooms can likewise be made to repulse each other (Arvan & House 2002). Even for simple buildings with few rooms, this approach will soon meet topological limits to the amount of possible connections in a two-dimensional or even three-dimensional architectural space. Some authors have tried to apply the attraction model to a hospital building typology, but the results are closer to a functional relation scheme then a building plan, at best this result will serve as input for a plan layout, see Figure 19 (Lorenz Bicher et al 2015).

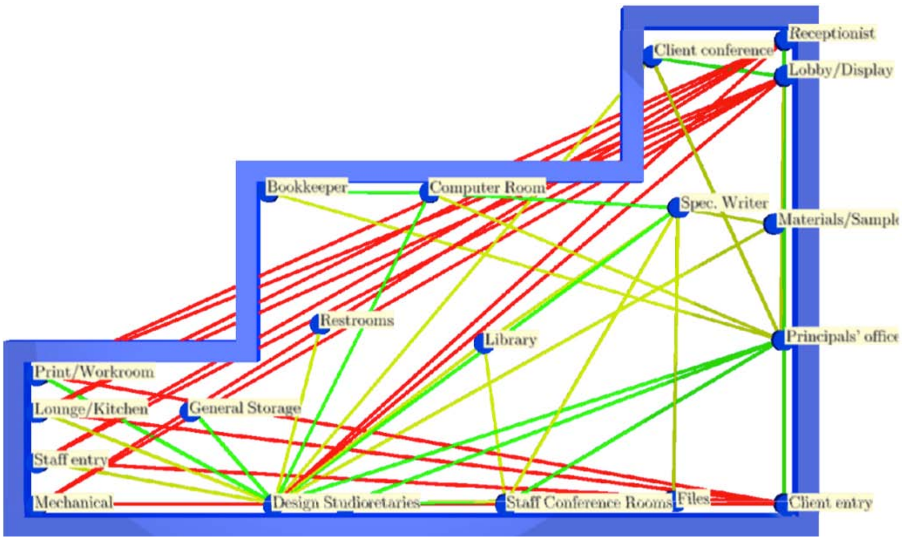

Figure 18: Outcome of physical simulation, from Lorenz et al 2015

The amount of solutions to a spatial problem increases exponentially with the amount of spaces, see Figure 19 (Liggett 1980, Gero & Jo 1998). Conventional linear calculation may not succeed in finding the best solution, it is more likely to find a local optimum.
Evolutionary design with optimization algorithms are an effective way to find a global optimum. It generates many alternatives, in many iterations.

In the number lies the chance of generating an optimal solution. Since the method is to me the most promising to solve layouts, Evolutionary Algorithms will be treated in the next chapter.

n	Number of solutions	n	Number of solutions
(Feasible to solve by hand)		(Feasible to solve by computer exhaustively)	
1	1	7	5040
2	2	8	40320
3	6	9	362880
4	24	10	3628800
5	120	11	39916800
6	720	12	479001600

Figure 19: Numbers of space elements 'n' and their possible solutions, Liggett, 1980

Tools

Project Akaba by Autodesk was an experimental software project allowing users to generate space layouts according to goals set by the user. Its option generation methods included: *"algorithm that scatters room "seeds" and encourages them to grow to the limit of the growth of adjoining spaces…a series of strategies for dividing up a known space…goal-seeking recursive optimizations using simulated annealing to find an acceptable space fit within a perimeter"* See Figure 20, on the left the underlying code to the design goal can be read and changed. As is clear from the generated designs on the right, external boundaries were pretty much ignored in project Akaba.

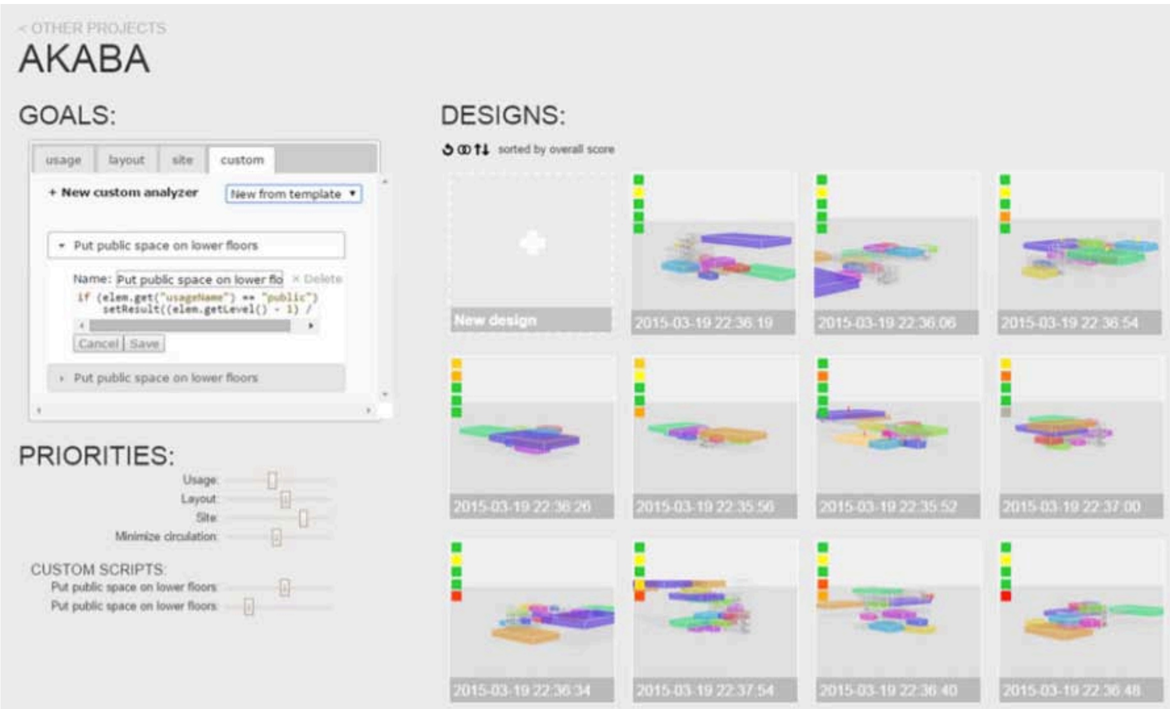

Figure 20: Project goals on the left, generated designs on the right in Project Akaba

The follow up to project Akaba is called project Fractal, it is being developed as an extension of the Dynamo visual programming interface of Autodesk Revit. It includes mass modelling options, automated core design, space relations according to the *linkography* method. A dynamic interface with sliders lets the user restrict the number of displayed design options. Progress on the development of the tool can be followed on the Buildinglab blog.

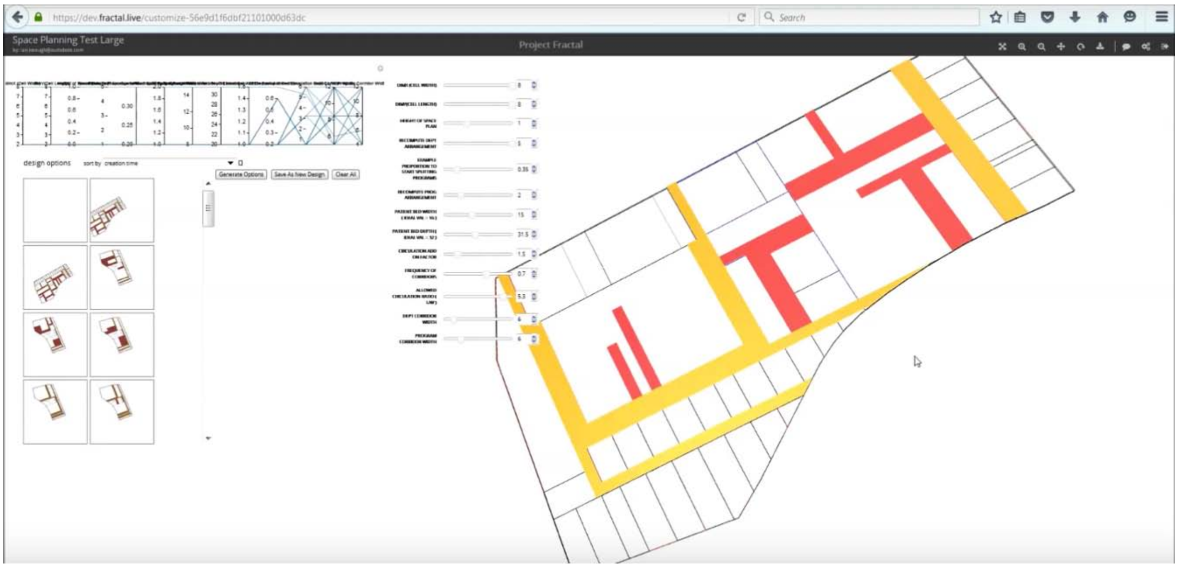

Figure 21: Plan layout in project Fractal, buildinglab

3.4 Evolutionary design

Evolutionary Algorithms have been used in programming since 1964 when A. Fogel published *"On the Organization of Intellect."* Outside of programming, the method gained visibility through "*The Blind Watchmaker"* by Richard Dawkins, a book about evolution. This came with a computer programme generating body shapes. See the Biomorphs video or the screen capture of a generation, Figure 22. Both book and programme serve to illustrate the point of the book that Darwinian evolution (or algorithms and selection) suffice to create intelligent design, the intervention of a conscious designer –or God– is not required as an explanation for life forms.

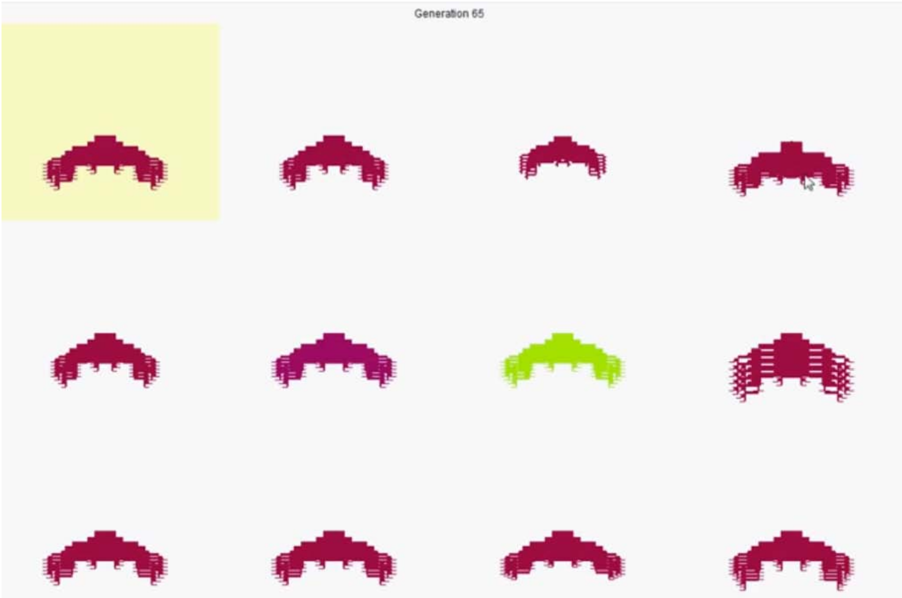

Figure 22: Biomorphs Applet, video screen capture of generation 65

Evolutionary algorithms use the metaphor of Darwinian evolution for calculating the most adapted solution. De Jong (2006) lists as basic elements:

- a population of “individuals”
- a notion of “fitness”
- a birth/death cycle biased by fitness
- a notion of “inheritance”

Parameters are interpreted as genes, recursive calculation cycles form generations. Individuals are defined by genomes or combinations of genes; successful genomes are more likely to be selected for the next propagation. It is a *blind* process, it first generates solutions then evaluates the solutions.

Popularity of evolutionary design has increased among designers due to the availability of the Galapagos evolutionary engine in the Grasshopper3d visual programming plugin for Rhino3d, first released in 2007. It does not require programming skills from designers that, as a group, are more visually oriented. The features and limits of evolutionary design algorithms have been clearly illustrated by David Rutten in the 2010 Advances in Architectural Geometry conference, and in a subsequent series of blog posts: Evolutionary Principles applied to Problem Solving.

He lists as disadvantages of the method its slowness, the exponential growth of possibilities for each additional parameter, and the high runtime cost of calculating all the options. Also, it does not guarantee arriving at a solution, or may not recognize the solution when found. As advantages on the other hand he lists the flexibility of evolutionary algorithms meaning that they will handle a wide range of problems. They are forgiving, as they will compute over- and under constrained problems. Furthermore, they are progressive, providing an infinite stream of answers, a characteristic which also allows for user interaction.

If two parameters are mapped on an x- and a y-axis, and their suitability on a z-axis, a Fitness landscape can be visualized, see Figure 23. The highest peak contains the best (fittest) solution. For the sake of the example to stay in a 3-dimensional space, only two parameters are mapped. Most real-world problems will require many parameters and consequently the global solution will be more elusive to calculate. This creates the need to generate many solutions, for many generations.

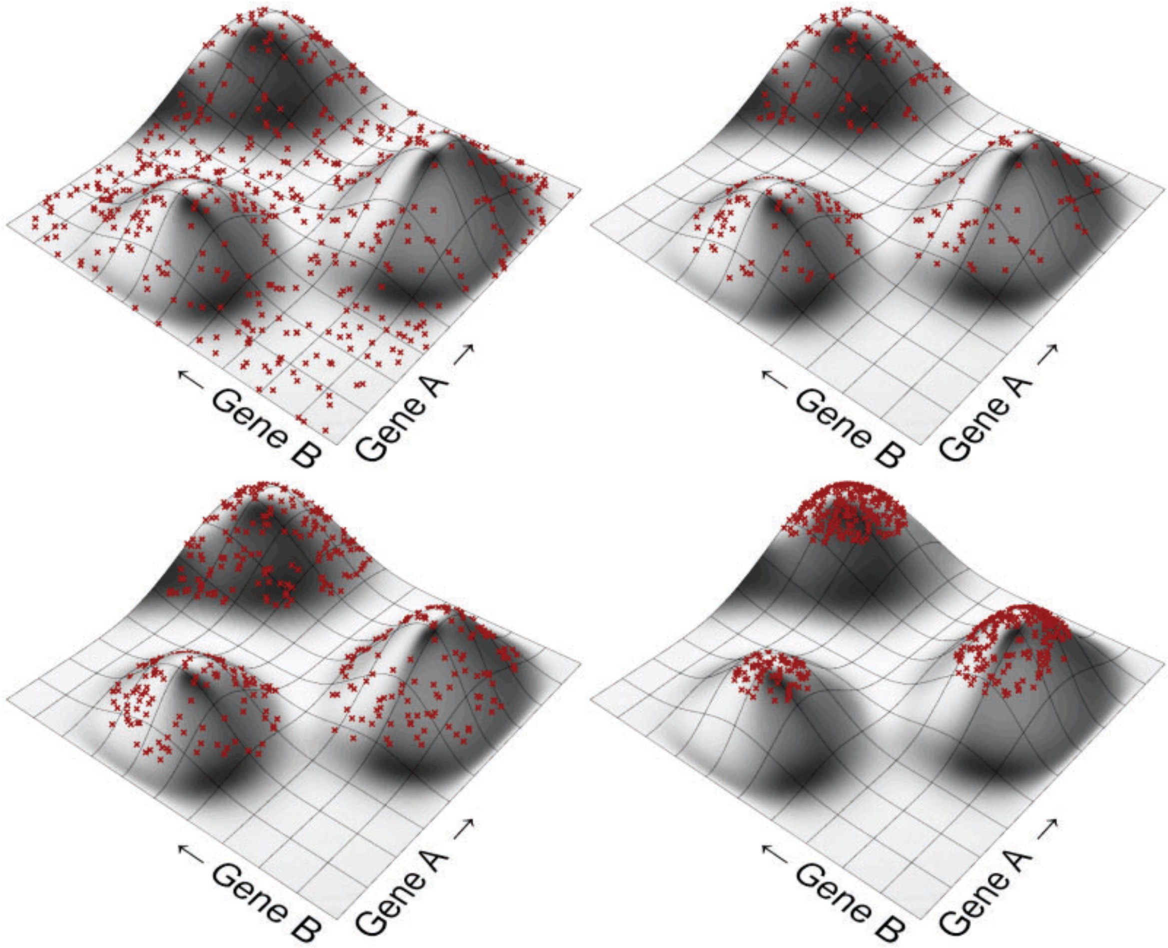

Figure 23: Fitness landscape with two genes, Rutten 2010

The four images show how the strength of numbers in evolutionary computing ideally finds the best solutions. Top left, the first population distributed randomly on the fitness landscape. Top right, the solutions with less fit genomes are eliminated. Bottom left, solution offspring from the previous population. Bottom right, after *n* generations the three optima are populated with the best solutions.

Rutten lists the components of an evolutionary solver as:

- Fitness Function
- Selection Mechanism
- Coupling Algorithm
- Coalescence Algorithm
- Mutation Factory

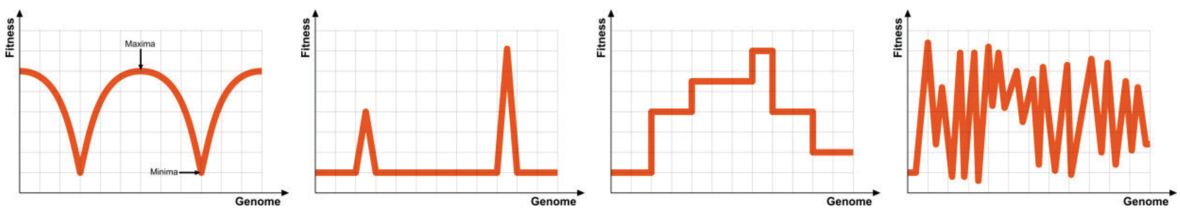

Figure 24: Fitness graph for single parameter (genome), Rutten2011

The four graphs in Figure 24 show different fitness functions, on the horizontal axis the genome, on the vertical axis the fitness. From left to the right the optimum solution is harder to establish by means of evolutionary algorithms, because of the plateau in the second and third image, and because of the random fitness in the last image on the right. With a non-linear or discontinuous fitness, generations may propagate on a local optimum and therefore fail to obtain the absolute optimum.
The selection mechanism refers to the process by which genomes get selected to reproduce, to generate offspring. Is it only the fittest genomes, all genomes, or a mix in which fitter genomes generate more offspring?
The coupling algorithm determines how much similarity there is between the genomes that get to reproduce. Coalescence algorithm refers to how the genes for offspring are determined from the parent genes, are they averaged or hard mixed from the parents? The use of chance factors applied to genes attempt to find solutions outside of the local optimum, this can be thought of as gene mutation.

Large part of difficulty lies in defining the optimum. Defining fitness well is crucial to the generated outcome. Some genes (parameters) may be pulling the solution in opposite directions, requiring non-linear responses. The acceptable solution space is therefore non-linear. Rutten gives two examples, of the optimization of a window size, and of calculating the division of a floor surface area that require non-linear approaches.

As an example in the field of layout design, think of two rooms that need to be adjacent or at a short walking distance. While locating them at ten meters apart will be fine, locating them fifty meters apart is not. Finding an appropriate layout solution, would accelerate if the design rules used to evaluate fitness take this non-linearity into account. Another non-linear example from literature is the quadratic assignment model which uses the desired distance between rooms to calculate the best solution (Burkard, Rainer E. *The Quadratic Assignment Problem* (1998) as cited in Elezkurtaj/Franck 2002).

Evolutionary algorithms may be the most successful strategy to generate plan layouts where there are many possible solutions, according to several authors (Elezkurtaj & Franck, Schneider, Fischer et al, chapter 3.3).
For complex buildings, such as hospitals, the PoR may be over constrained. An evolutionary solver will calculate solutions. Fitness will have to be established by well-chosen design rules.
Often however, in architecture the design question is poorly defined: For some buildings, the Programme of Requirements (PoR) is given as a mere list of spaces with areas but containing little more information, the capacity of the evolutionary solver to deal with under-constrained problems should allow calculating layouts. Again, the definition of fitness is crucial.

To allow the designer to deal with poorly defined design problems, Schneider, Fischer et al propose a method in which the designer can interact with the calculation on different scale levels; e.g. the urban block, the building floorplan, the apartment floorplan. It is the evolutionary solver that allows this kind of interaction, because it can visualise the best available solution(s) at each point in time.

Tools

As already mentioned, a relatively accessible and widely used tool for applying evolutionary design algorithms is the Galapagos evolutionary engine in the Grasshopper3d visual programming plugin for Rhino3d modeller. Many examples of its use can be found through the dedicated group Galapagos. Its flexibility and accessibility comes from the fact that it works alongside all the different components in Grasshopper3d ecosystem. Any of the parameters used in the Grasshopper3d environment can be used in the evolutionary solvers, so any combination of parameters can be simulated and optimized. The art lies in defining the solver problem, within the conditions listed above. Grasshopper3d includes components that simulate physics (Kangaroo3d), that model structural analysis (Karamba3d, Millipede), that model structural elements to IFC4 (GeometryGym), that perform daylight and energy analysis (Diva, Ladybug, Geco), and also that provide functional relation diagrams for spaces (space syntax).

Octopus is another evolutionary solver within grasshopper3d, it adds specific controls on the solver settings.

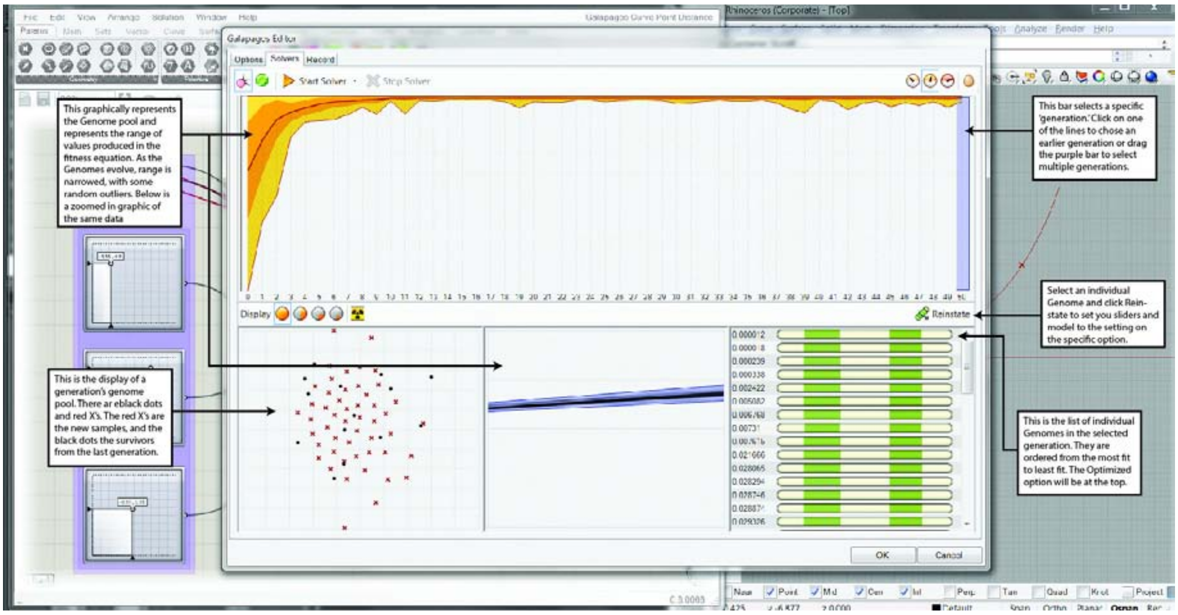

Figure 25: Galapagos Editor, from UMN Digital Design tutorial

3.5 BIM to Early stage building energy simulation

Energy simulation in early stage design has been identified as a priority for BIM interoperability processes. A great deal of information in architects' building models could be used for energy analysis purposes. The objective is to reduce the manual operations from architects to energy simulation model, reducing simulation time and cost for each design iteration. An early feedback on a design proposal is more likely to have a positive effect on the next design iteration.

Looking for an open process, in which the BIM modelling tool and the energy simulation tool are not fixed *a priori*, an exchange file format is required. Candidates for the exchange from the BIM model are the IFC file format and the gbXML format. The IFC file has many different possibilities for defining the object geometry, e.g. extrusions, Boolean operations. The energy analysis typically only works with meshes or planar geometry. The geometry transfer needs to be controlled. Also, the IFC file may contain information unverified from the architect such as the default properties of library objects, or information may simply be superfluous to the Energy analysis (Marsh2006).

The gbXML format was developed by Green Building Studio for the sole purpose of energy modelling. The export from the BIM model is therefore more likely to contain the right geometry and information, spaces, zones, surfaces and materials are clearly defined. On the downside on export it loses the building orientation axis (ibid.).
The first successful transfer from IFC to energy analysis software was developed in 2001 by MagiCAD's Olof Granlund under the name RIUSKA. On this basis BSPro application was developed by 2011.
A significant multidisciplinary effort to improve the IFC based information transfer was undertaken in the 2009 AECOO Testbed project led by the Open Geospatial Consortium. The project produced Building Performance and Energy Analysis (BPEA) and Quantity Take-off MVD's.
An Archicad BIM model was exported to IFC according to the BPEA Model View, an MVD which was itself defined within the AECOO Testbed project. In the next step, the model was checked for completeness in Solibri Model Checker.

While BIM element models represent a wall as a volume, energy simulation models consider a plane without thickness. These simulations must make different calculations for parts of the same wall if facing another space (possibly with another room temperature) on the other side of the wall, see Figure 26. As the same elaboration needs to be applied for floors, the amount of space boundaries can exceed the amount of physical building elements by many times.
In the words of the authors of the Space Boundary Implementation Guideline: "*Space boundaries are virtual objects used to calculate quantities for various forms of analysis related to spaces or rooms in buildings."* Analyses that use space boundaries include: Energy Analysis, Quantity Take-off, Facilities management (Weise2009)

Second level Space boundaries were elaborated with a specially developed Geometry Simplification Tool (LBNL, Bazjanic) converting the IFC file to the Energyplus file format IDF.

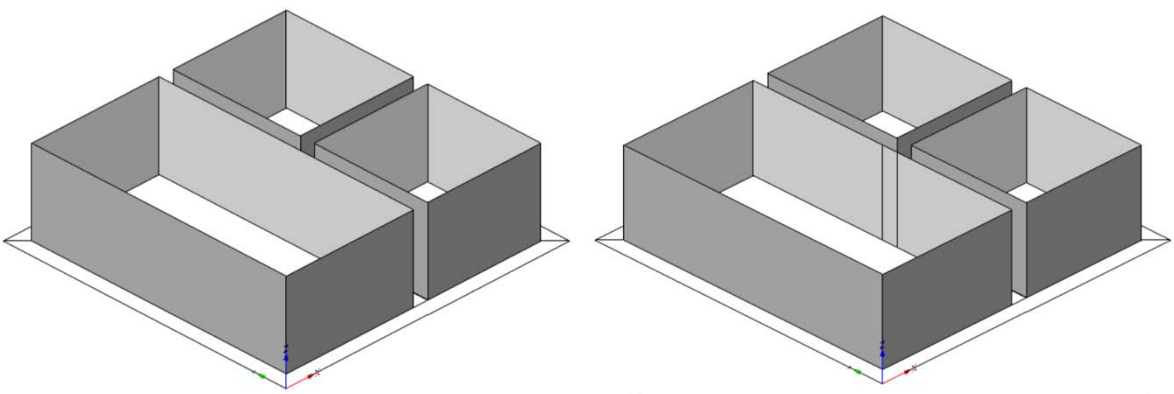

Figure 26: First level and second level space boundaries, from Hitchcock et al 2011

From these experiments, Richard See/Digital Alchemy proceeded to develop Simergy Pro which is the first EnergyPlus interface built to directly interpret the IFC file. [2]
For an optimum automated exchange, ideally the simulation tool can interpret the IFC file without manual input. A project tailored workflow will be illustrated in paragraph 4.1.9, KPI: Evaluating energy performance.

Tools
Space Boundary Tool is the follow up of the LNBL Geometry Simplification Tool, it is "*a tool for automatically calculating whole-building energy performance simulation thermal space boundaries for Industry Foundation Classes (IFC) building models, and using these space boundaries to automatically generate EnergyPlus Input Definition Files (IDFs).*"

IFC compatible simulations tools include Simergy Pro based on the Energyplus engine, Equa's IDA-ICE, VABI Elements. Some of these programmes are calibrated to national regulations and as such may not satisfy the criteria of all projects.

Other early stage simulation tools include Designbuilder and Autodesk Ecotect (latest release 2011), integrated into modelling software are Eco Designer Star within Archicad by Graphisoft, Solemma's DIVA in Rhino3d and Sefeira integrated in Sketchup and Revit.

3.6 From evidence based design to performance indicators

From evidence based medicine derives evidence based design, an approach developed in relation to healthcare design where differences in design solutions may directly influence patience wellbeing. The method attempts to eliminate the arbitrary, personal preferences of designers in favour of proven design solutions.

Cama (2009) defines Evidence based design as an iterative decision making process based on evidence, it works with behavioural, organizational or economical clues with a design objective. It provides a knowledge platform rather than prescriptive solutions. It measures outcomes and shares results in a peer-reviewed manner.
In the same book Cama lists four basic components of evidence based design:

1. Gather quantitative and qualitative intelligence
2. Map strategic, cultural and research goals
3. Hypothesize outcomes, innovate and implement translational design
4. Measure and share outcomes

Once the outcomes have been established, it becomes possible to establish new criteria which design proposals must meet. These criteria are known as Performance Indicators, or when weighed and bundled, Key Performance Indicators. The application of Performance Indicators, grouped as Key Performance Indicators (KPI's, see paragraph 4.1.8, p.74)
Performance based design evaluates and selects designs according to scientifically proven objectives, this meets the needs of decision makers, the public, and tax payers.

[2] This premise inspired the Italian case study of the Streamer project to apply a development version of Simergy Pro, however encountered problems e.g. the used versions' impossibility to interpret an inclined roof, forced switching to another, less direct workflow.

4 Semantic BIM design methodology

4.1 New hospital building in the Streamer workflow

4.1.1 Why hospitals?

Hospitals are among the most energy consuming building typologies. An average hospital district in a medium size European city use as much energy as 20.000 households; an average hospital building consumes as much as 2,5 times energy as an office the same size. The EU has around 15.000 hospitals.

Looking at the major trends for hospitals, we can determine three distinct drivers that possibly influence the demand for (more) energy (DiGiulio2015, Streamer D1.3):

1. More intense usage of spaces (more treatments per bed during prolonged periods of the day) -> impact on electricity use;

2. An increased demand for comfort (from partially air-conditioned to fully air-conditioned buildings, impacts specifically on ventilation);

3. More intense diagnostic & treatment possibilities and demands for these heavier treatments in hospitals (as lighter cases are treated at other places) -> expected impact on electricity use

Hospitals also need to meet strict functionally complex programmes of requirements, indicating large amounts of specific space types, and space relations the design must satisfy. The design of a hospital needs to take many factors into account. The automated generations of design proposals is likely to be of assistance to the designer. However, in a preliminary phase exact values may not be available or a high level of accuracy may be undesired. A solution to this is the adoption of classification, grouping spaces according to similar functional requirements based on prior experience of design processes. (Streamer D1.3)

The Semantic BIM design methodology described in the following section tries to leverage the available data in each step of the design. The choice of the IFC file for the building model helps preventing loss of data as it collects all the outputs. The brief of requested spaces and relations is used to generate design alternatives. The room requirements contained in the file inform the energy simulation and all the collected data can be enriched and validated by means of reasoning. Finally, a *dashboard* shows decision makers how the alternatives compare. The IFC file of the chosen design can then be imported in a BIM modelling application for design development and onwards.

4.1.2 Brief description of semantic design method.

Before proceeding with a detailed description, a short summary of the Semantic BIM design process is in place: The brief is analysed and all spaces are categorized, concerning functional and technical requirements. Subsequently, design rules are defined which will use the semantic information contained in the brief. Boundary conditions such as site, building volume and orientation are inserted in an Early Design Configurator (EDC) application. Using an evolutionary algorithm, the application then generates series of spatial layouts, which are scored according to weighing factors the user assigns to the design rules. The highest scoring options are memorized, at any given time the highest scoring designs can be exported as an IFC model. A manual intervention is the choice of building envelope and MEP systems, supported by filter rules. The designs are checked against semantic rules in the Early Design Validator (EDV).
The models are then analysed for their performance in the areas of energy (in the TNO Energy Calculation Tool, or TECT), cost and quality. To facilitate the comparison,\ a decision support tool (DST) may be used, which illustrates the results synthetically and allows for further calibration.

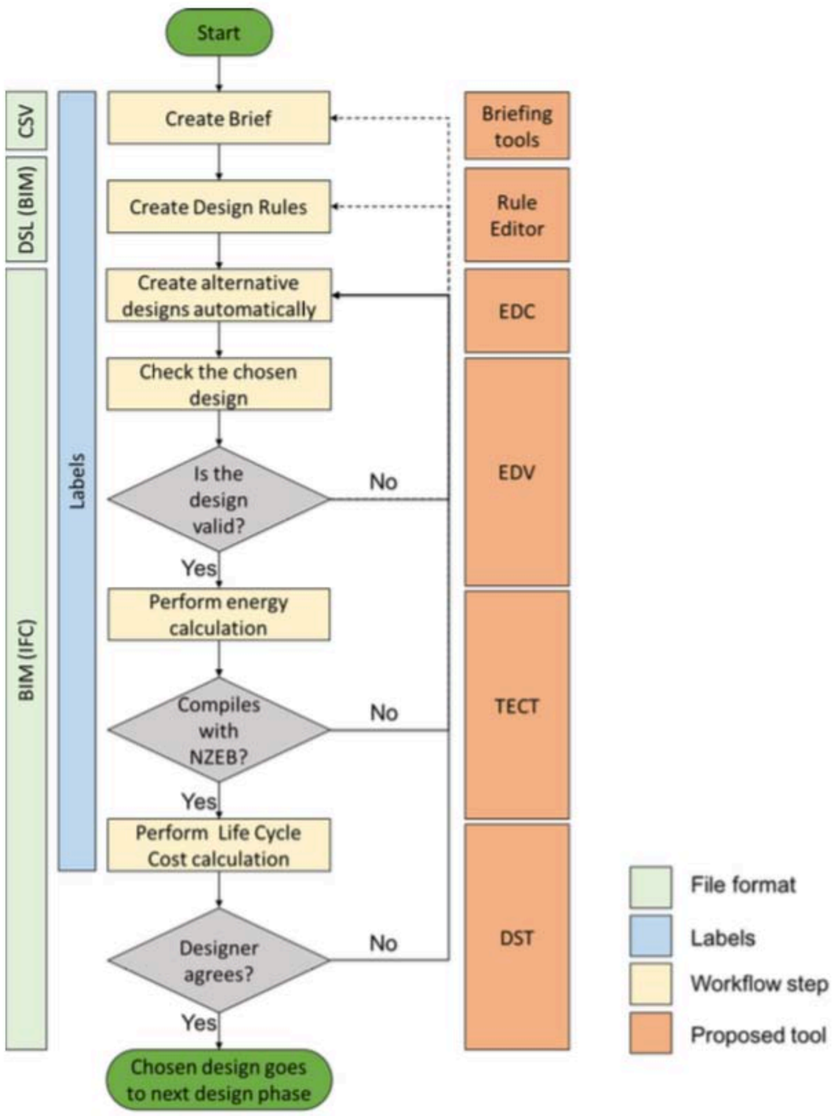

Figure 27: Proposed Early Design Workflow, from Sleiman et al 2017

4.1.3 Classifying the brief

Manual constraints, bottom-up input

In an analytical phase the Streamer project distinguishes four scale levels: district, building, functional area, room (De Hoogh2014.)

- District level; a number of health related and supporting buildings make up a healthcare district
- Building level; technical, functional and volumetric unit
- Functional Area level; collection of functionally related spaces within the building
- Room level; smallest spatial unit, functional and technical requirements may vary at room level.

Sometimes properties will be inherited from a higher level. For instance, all rooms will share certain properties, based on the building they are located in. Other times properties will be dictated by a lower level, e.g. operating theatres may require higher ceilings and floor loads which will then apply to a larger area.

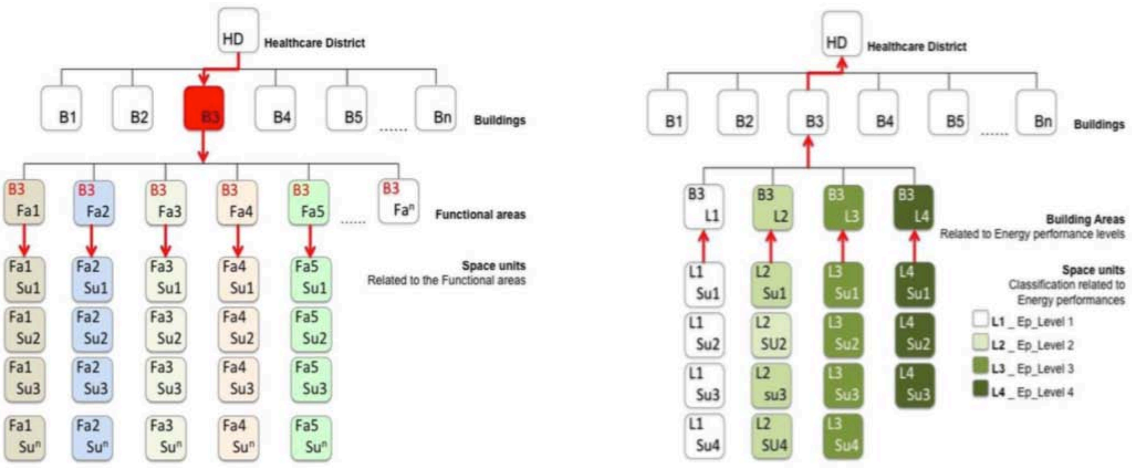

Figure 28: Top Down (left) versus Bottom Up (right), Streamer D1.6

Top Down versus Bottom Up

Design of complex building programmes generally requires designers to think in groupings of spaces with related functions, in the hospital context called functional areas. These groupings may coincide with hospital departments or may be subdivided into finer functional or organizational groupings. A design method which attempts to compose at this scale level may be described as a Top-Down approach. An architect might start by aggregating the required spaces in Functional Areas, then design a schematic layout at a scale level in between the building level and the room level. However, this conceptual simplification temporarily ignores all the technical aspects at the room level. These are postponed being dealt with later, on a lower scale level. Constraints related to the site, the building and the clustering of functional areas can be defined as a top-down approach.

In a parallel process, a more technical approach might try to aggregate rooms, based on technical requirements of individual rooms. The technical requirements of the single room would influence its placement thereby influencing the larger whole. This is a bottom-up approach, using information on a room level, or even on its medical equipment. Therefore, it obliges us to collect technical requirements on each individual room.

One common design method in a traditional design process is to alternate Top-Down and Bottom-Up approaches, at one moment considering the functional organization of the entire hospital building, at the next zooming in to room level. While letting each scale level influence choices on the other scale level, logical or intuitional mental steps further the design in a recursive process.

In the case of the EDC the simultaneous application of design rules will instead have to substitute this alternation. The computer will try to generate a layout based on all the input including the PoR and the Design Rules. A design rule which tries to group spaces belonging to the same functional area (equivalent to the Top-Down approach), is formulated using a previously assigned a Functional Area label to each required room in the PoR. These rules work to control the functional aspects.
From the opposite scale level, the room labels such as Construction, Hygiene, Use class may be used (equivalent to the Bottom-Up approach) to obtain groupings of similar technical requirements. These rules hope to obtain an optimization of resources spent in construction and maintenance. Also at a smaller scale level, but with a functional aspect lie the rules defining a Room-To-Room relation.

Within the proposed method, all the higher level, top-down choices regarding the building, such as the composition of its volume, depth, corridor placement are made by the designer during the manual definition of the building volume and its constraints. The architect remains very much in control of the design. Because these choices are part of the expert user’s knowledge, no labels are used, just as in a traditional design process.

The specifics of the design and evaluation method define the limitations or the set of possible solutions that may be found. Ventilation, heating and cooling demand, (day)lighting and appliances respectively, are the factors which have most impact on energy demand. All these factors except for appliances are influenced by the building depth, a top-down constraint that is set manually. Hot floor and Industry labels are generally expected to require a deep plan. Hotel and office functions typically are suitable for shallow plans.

Within the manually defined constraints The EDC application will try fitting the PoR required spaces. While the distinction between top-down and bottom-up can be clearly distinguished, the EDC can apply both simultaneously. How it evaluates the constraints, will depend on the evolutionary algorithm, the project specific design rules and the rules’ relative weight.

Table 7: Top-down & bottom-up constraints

Top-down constraints (site-building-FA)	Bottom-up constraints (PoR & Design rules)
• Site (building access)	• Level of access restrictions
• Building envelope (depth)	• Spatial & functional type
• Corridors	• Comfort level
• Vertical elements (shafts, stairs)	• Floor height
• Grids	• Cleanliness level
• Functional Area aggregation	• Occupation schedule

Classifying the rooms

For the bottom-up approach we will need to classify each room and describe the requirements of each room type. In preparation for the early phase design, accurate information on systems and components is not available in the early design phase. Instead, based on previous hospital buildings and often prescribed in national regulations, room types and their properties are largely classified. While facing the design of complex buildings with elaborate requirements as are hospitals, the reduction that comes with classification helps to consider aggregations.

Classification systems have long been in existence, examples that name and classify rooms include Omniclass, Uniclass. These systems help organize available information but do not define the actual room requirements.

More recently definition of the PoR is assisted by web based applications (dRofus, Briefbuilder) that allow room types and space requirements to be defined. These applications come with a direct link to BIM modelling applications (e.g. Revit, Archicad, Allplan), which allows the room requirements to be compared with the design for the duration of the project. The specification can be at a very detailed level including all the equipment requirements for each room (Traversari2017labels).

An approach used in Dutch healthcare design groups hospital spaces in four different spatial and functional typologies referred to as “layers”. This refers to a quite a coarse grouping of building types. Every hospital function can be thought of to require a spaces and systems similar to a hotel, an office, a highly technical building (*hot floor*) or an industrial building, see below. While conceptually clear, it is not descriptive enough to inform the energy simulation or to organize the connections at the room level.

The Semantic BIM design method simplifies and structures the room level requirements in the PoR. A selection is made of properties required in early design phase. The selected properties are grouped into classes that share similar values. These properties are then aggregated and translated into *semantic labels*. As the labels are stored as room requirements in the IFC file (PropertyRequirement in ifcSpace), they can be used for inferring other properties in a later phase. The labels thus obtain a meaning that is wider than the strict initial definition.

The first phase of the semantic design project is thus a selection of concepts that need to be considered. The criteria for selection can be defined as:

- Factors that have a strong link to functionality and determine topology/layout
- Factors that influence energy consumption
- Factors that influence quality and cost

Within these factors the most important to consider early on, are the ones that are harder to change as the design will be further developed. These factors will include building volume, plan depth, circulation, distribution of functional areas. This list of factors that are difficult to change can be summarized as geometry and topology.

Some of the selected factors may be bundled conceptually, as high performance requirements will frequently be shared by some space types, as well as low performance requirements by others. For example, under the comfort level label we may group climatic comfort (air flow rate, air exchange rate, temperature range), visual comfort (lighting levels, daylighting, glare control) and acoustic comfort.

The term *semantic label* refers to the requirements which are attached as information to the PoR, in which the grouping allows for a conceptual simplification. Having established the need to simplify by classifying and categorizing, let us look at the selection of label requirements.

Bouwcollege layers approach

Hospitals are expensive buildings to design, build and maintain. The requirements of some functions within the building may be very high, for example the floor loads for heavy medical equipment, high ventilation rates, deep ceilings resulting in high floors. One attempt to reduce investment cost and increase leasability was laid out by the Netherlands Board for Healthcare Institutions (2007) in: *the Building Differentiation of Hospitals; the Layers approach*[3]. In short, this approach presumes efficiency of means by building specifically adapted typologies, it goes on to claim these buildings may be used for profit when they are no longer needed for public hospital functions.

[3] The term layers may be a mistranslation from the Dutch word for floor levels (*bouwlagen*), since each layer is determined to have a specified floor level height. The NBHI document mentions placing spaces on a “layer” which makes sense only if interpreted as a floor level. If anything, this error shows the need for the formally agreed concepts in the bSDD.

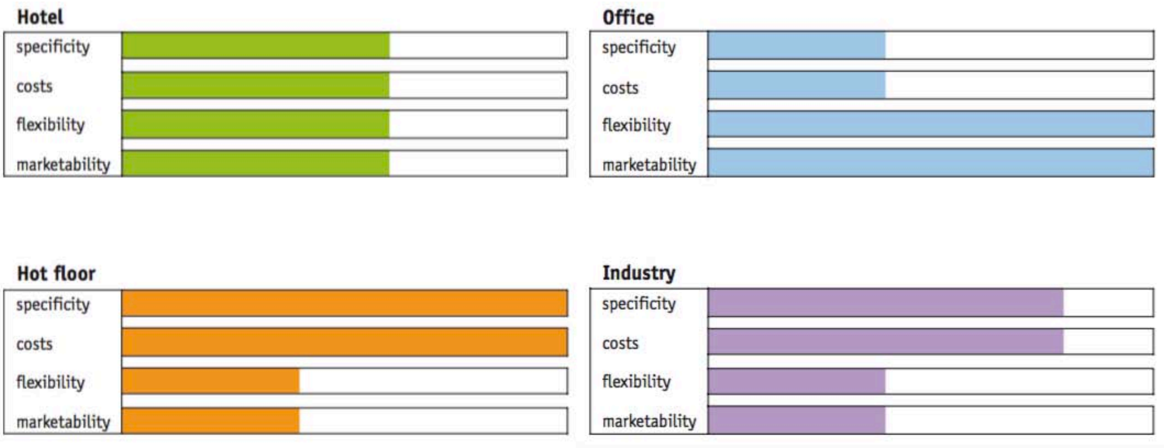

Figure 29: Properties of a hospital's layers, from NBHI207

The main categories assigned are a grouping by typology; this presents a design strategy that derives from a market based approach to investment in hospitals. Real estate is considered *"… no longer a given, but a means of production contributing to efficient business operations in healthcare, where integral funding takes over from a separate financing flow for construction."*

"The layers approach divides the hospital into four buildings, referred to as the layers. The first layer, the hot floor, comprises the high-tech, capital intensive functions that are specific for hospitals. The hotel comprises all functions for accommodation of patients. The functions for diagnostics and simple examinations and treatments are accommodated in the office. Logically, the office also accommodates the office facilities, such as staff accommodation, accounting and management. Last but not least, industry accommodates all medical supporting and facilitating functions."

According to the method, all hospital functions can be classified as belonging to one of the following building types:

- Hot floor for hi-tech capital intensive specific hospital functions
- Hotel; e.g. patient accommodation
- Office; e.g. day hospital, polyclinic
- Industry; e.g. kitchen, laundry

The NBHI study compares three conceptual models; the monolith, a single building containing all functions; with a hybrid and an extreme model, a separating the layers into typologically different specific buildings (Figure 31). Further development has seen intermediate solutions, for instance stacking the different hospital functions on top of each other in building levels.

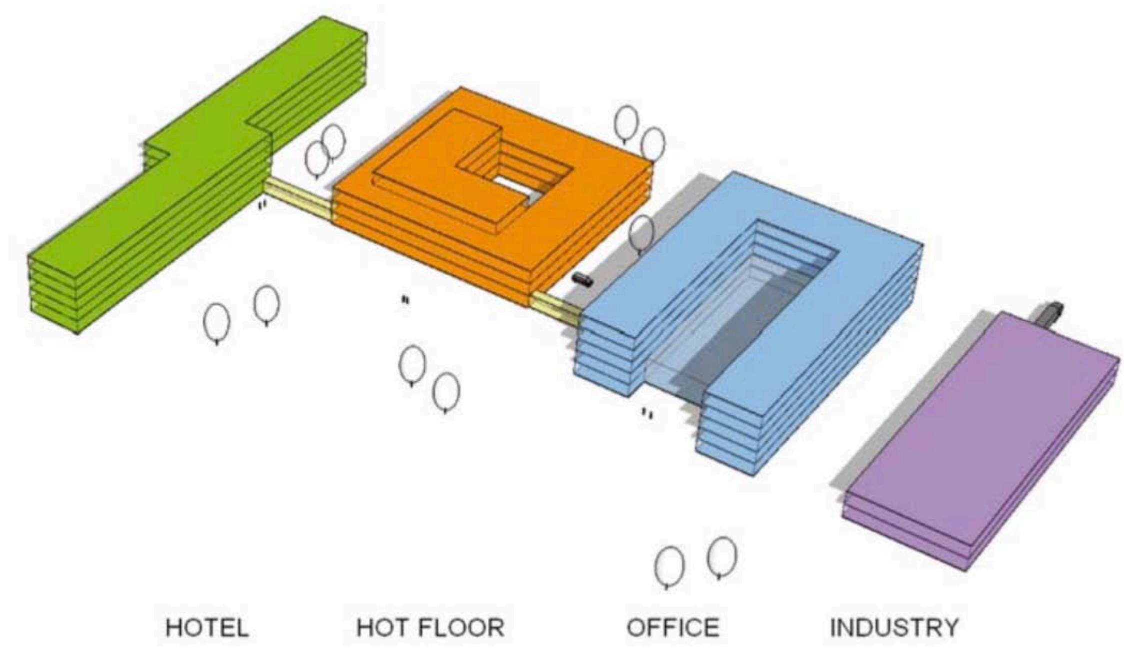

Figure 30: Perspective Drawing of the Extreme Model (NBHI 2007)

The listed advantages include reduction of investment cost due to lower building cost, each function is tailored to its own minimum requirement (ceiling height, floor load, daylight access).
On the downside, the differentiation in performance requirements will limit the future possibilities of shifting functions around between buildings with different layers, as fewer buildings and floor levels will be designed with an over capacity. This may pose a limit to flexibility.

The Streamer project uses the named categories given its suitability for broadly classifying spaces in the Early Design phase. The actual importance given to the *Bouwcollege* layer whilst generating designs may be determined in a later step (paragraph 4.1.4).

Room types & functional areas

For the sake of oversight, the requested rooms are reduced to a limited number of room types, each of which is assigned with several default properties. Each room type must be assigned to a Functional Area. Each room type also has a default Bouwcollege label.

Distribution spaces are a special room type as they are not generally requested in the PoR but rather must be added as within each generated plan configuration based on the requested connections between the rooms. An informed energy simulation requires that corridors have the same property sets as do the other spaces. If the values are assigned by default, subsequent checks will need to be performed later in the process.

Selecting semantic labels

Since functional requirements are essential for hospital buildings, some essential properties need to be selected as indispensable for the Early design process.
An overview of the final Room label properties in the Streamer process is given in Table 8: Room label properties and allowed values in the Streamer project.

Table 8: Room label properties and allowed values in the Streamer project

Property label	Description	Allowed values
AccessSecurity	Relation to level of access restrictions to staff or public, building envelope	A1, A2, A3, A4, A5
BouwcollegeLayer	Room spatial & functional type according to Netherlands Board for Healthcare Institutions	H (hotel), O (office), I (industry), HF (hot floor)
ComfortClass	Relation to building envelope, air flow, temperature range, lighting, relative humidity, indoor noise levels and control of lighting	Ct1, Ct2, Ct3, Ct4, Ct5, Ct6, Ct7, Ct8
Construction	Relation to construction typology, floor height, floor strength, shielding against radiation	C1, C2, C3, C4, C5, C6, C7
Equipment	Relation to electrical power usage, medical gasses and emergency energy supply	EQ1, EQ2, EQ3, EQ4, EQ5, EQ6, EQ7
HygienicClass	Relation to required cleanliness, ventilation factor, air tightness, materials and windows	H1, H2, H3, H4, H5
UserProfile	Relation to opening hours and room occupation schedule	U1, U2, U3, U4, U5

The column headed "Description" in the above table clarifies how each semantic label will determine parameters across three areas:

1) parameters required for the Early Design Configurator;

2) parameters for the selection of a Mechanical, Electrical and Plumbing (MEP) system;

3) parameters for the selection of an Energy efficient Building (EeB) solution.

Table 9: Streamer labels for different Bouwcollege layers

LAYER	DESCRIPTION	LABEL VALUES
HOT FLOOR	Accommodates the high-tech, capital intensive functions that are specific for hospitals e.g. operating theatres, isolation rooms, emergency	H4, A5, U4, EQ4, CT7, C3
HOTEL	Comprises all functions for accommodation of patients e.g. patient rooms, general nursing, day nursing,	H3, A2, U4, EQ3, CT4, C1
OFFICE	Accommodates the office facilities e.g. staff accommodation, accounting and management	H2, A4, U1, EQ2, CT3, C1
INDUSTRY	Accommodates all medical supporting and facilitating functions e.g. production pharmacy, laboratories, imaging centre	H5, A5, U3, EQ6, CT6, C4

"Although the semantic labels at space unit's level are developed for rooms, these labels can also have values when applied to functional areas. The philosophy behind the application of semantic labels is identical at both levels. If no independent information regarding properties at the functional areas levels… … is available, the semantic labels for the space units can be used to infer these in the early design stage." (Traversari2017label)

Reduction of the building model to a concept (mvdXML)

The Streamer semantic baseline design model in the Early Design Stage includes the following elements:

- Objects: Building (defined in EDC and part of IFC export), spatial units (defined in PoR and exported in EDC in IFC export) and components (part of export of EDC as IFC, examples are: floors, walls, doors, windows etc.).
- Attributes: labels, room typologies, functional area typologies, building compactness ratio, simulation results, spatial information
- Relationships: IFC structure, geographical locations and indirect by means of design rules.

The consequence of reduction to space types with few required properties, according to classes described above, is that the exchange requirement is equally synthetic.
Figure 33 shows the schema of required information that the EDC shall deliver. The IFC model is reduced to spaces, with some labels and basic quantities.

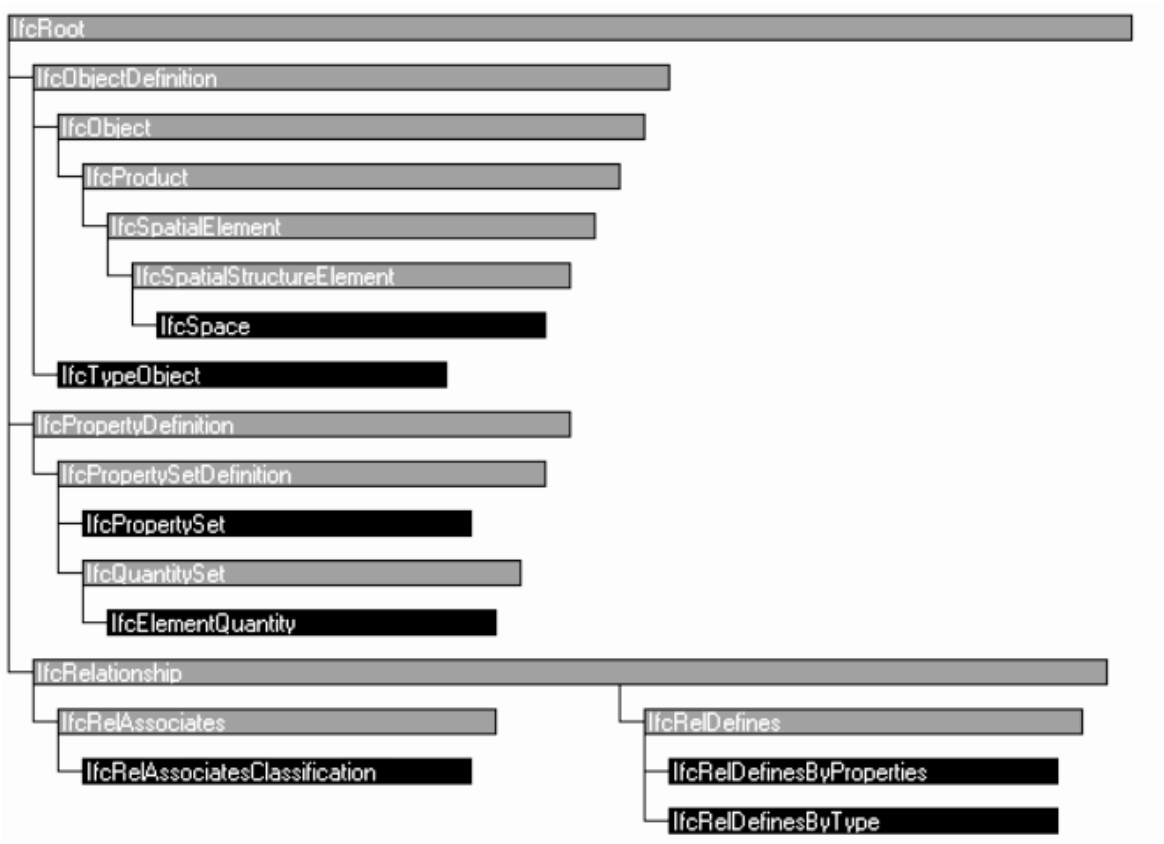

Figure 31: Schema of the architect/EDC mvdXML requirement, in the IfcDoc Tool

Tools

Uniclass2, Online searchable classification maintained by the British based Construction Project Information Committee. A total of 58 Medical, Health and Welfare space classes are preceded by code Sp_35.

Omniclass, Downloadable tabular format maintained by the US based Construction Specifications Institute. Finer grained than Uniclass, it contains a total of 436 Healthcare space classes that are preceded by code 13-51.

Commercial PoR editors including dRofus /briefbuilder share the following properties:

- Multiple users can add and modify the PoR, via a web service or webpage
- A plugin provides access to the database from inside the BIM modelling application
- Synchronisation with the modelling environment ensures consistency even when the PoR is updated during the design process
- Most data is room-based, and is recorded as parameter value to rooms in the modelling environment
- A tree-structure ensures rooms belong to functional areas and functional areas belong to buildings
- The user license is project-based and usually paid for by the client

dRofus, requirement database with BIM link, documents systems and equipment as well.

Briefbuilder

In absence of a dedicated PoR editor, to select Streamer labels, a spreadsheet can be used to select rooms and properties from a list of predefined values. The required output is a CSV file which should contain only agreed values and is case sensitive. As a CSV file does not have a standard structure, an alternative would be to translate the PoR into an IFC file which can then be checked against a Model View with an mvdXML.

4.1.4 Defining design rules from desired relations and from building code

Influence spatial relations with semantic labels

Once the spaces have been defined in the PoR and their individual property requirements have been assigned in the form of semantic labels, the same labels can be used to manipulate the spaces according to design rules. The design rules define desired relations between rooms based on the room type or any of the semantic labels, or between rooms and other building elements, such as the stairwell, or the façade.

Design rules can be formulated from existing domain knowledge (Streamer D1.6):

- Functional relations (e.g. departments and knowledge fields);
- Safety considerations (e.g. fire escape routes);
- Expert knowledge (e.g. staff operational efficiency, patient well-being, energy efficiency, building service efficiency and health risks);
- Best practices;
- Other experiences, preferences, legalisation and acknowledgements.

In hospital design, the higher priority will generally be assigned to the functional requirements. While the method proposed by the NBHI assumes that functional areas can be separated based on their *Bouwcollege* layer (Hot Floor, Hotel, Industry or Office). It does recognize how some functions need to be situated in proximity or with a direct unhindered connection. Two types of relations are distinguished according to the method proposed by the NBHI.

Medical relations between functional areas that need is needed to guarantee quality healthcare. The walking distance is important so close proximity or a direct connection are required, examples given are:

- emergency and operating theatres
- emergency and diagnostic imaging
- emergency and coronary care
- emergency and intensive care
- operating theatres and intensive care
- operating theatres and delivery

Organizational relations are not required from a strictly medical reason. Frequent contacts define strong organizational relationship which creates a need for proximity. The given examples are:

- paediatric nursing (neonatology) and operating theatres
- paediatric nursing and maternity
- maternity and delivery
- (surgical) day nursing and operating theatres
- laboratory clinical chemistry and emergency
- laboratory clinical chemistry and intensive care

So firstly, there may be strict functional, or organizational requirements regarding the adjacency of spaces, e.g. generally spaces that belong to the same Functional Area shall be clustered, horizontally or vertically, or as a specific example the operating theatre shall close to x-ray department. The labels to consider are the Functional Area label, AccessSecurity label, and RoomType.

For some examples of design rules at a functional level. One can think of clustering by Functional Area label, this is very similar to the architects' traditional top-down approach, in which the first, higher level grouping of required spaces is based on functional similarity. Clustering by BouwcollegeLayer will optimize the building typologies, and consequently reduce investment cost as argued by the NBHI layer approach. Lastly desired connections will be based on individual room to room relations.

An entirely different set of relations can be based not on functional relations, but on technical similarities. This follows the engineering bottom-up approach. These requirements are less strict than functional relations, but possibly of consequence on energy consumption. For one, expected optimizations may be based on the clustering of spaces with similar temperature, thus reducing thermal transmission losses between spaces. (as indicated by the ComfortClass label).

Clustering spaces in the spatial layout per scheduled usage, would make it possible to shut down, when not in use, entire areas with similar active hours (UserProfile label). The potential reduction in energy consumption is significant.[4]
Grouping similar MEP systems thus reducing system length, will be possible by clustering spaces with similar systems horizontally and vertically (uses the Equipment label and the optional MEP and lighting label). This will be more significant for spaces with high ventilation volumes or high electricity consumption.

For sample design rules, see Table 10. The first rule ensures all spaces that require daylight are placed along the exterior wall of the building. The second rule clusters spaces that have similar active hours (using the UserProfile label). The third rule attempts to place the admission space on the lowest floor level. The fourth rule uses the desired functional area to group spaces. The fifth rule defines a room to room requirement. To prioritize the rules a number indicates the importance attached to each rule. This allows a score to be calculated for each design, the highest scoring designs are kept in the memory. The number nine, the highest priority number indicates that the rule must be satisfied by the design.

[4] An analysis of TNO has shown that over a third of energy consumption in a hospital building occurs at a time *when no occupants are present*. This is due to lights and equipment that are not switched off, and ventilation and climate installations that remain active, see Figure 32. Source: Plugwise, ErasmusMC 2012, Streamer D1.3

Table 10: Example of rules implemented in the Domain Specific Language (DSL)

Design rules
priority = 9 Rule “Placement of all rooms requiring direct daylight and view outside”: Space with (ComfortClass equals “Ct1”) must have horizontal separation of 0 meters to exterior wall;
priority = 5 Rule “"Grouping of rooms with similar user profile values"”: Space with (UserProfile equals “U2”) must be clustered horizontally and vertically;
priority = 9 Rule “Admission story rule”: Functional area with (name equals “Admission”) must be contained in the lowest story;
priority = 8 Rule “Functional Area clustering rule”: functional area with (name equals “LowCareWard”) must be clustered horizontally and vertically;
priority = 5 Rule “Traveling distance between PatientRoom and NursingStation”: Traveling distance between space with (name equals “PatientRoom”) and space with (name equals “NursingStation”) is less than 20.0 m;

Gemiddeld uurverbruik per dag normweek (kW)

Uur / Dag	ma	di	wo	do	vr	za	zo	Uurtotaal (kWh)
0:00	1.5	1.8	2.4	1.9	1.8	1.5	1.5	12
1:00	1.5	1.8	2.4	1.9	1.7	1.5	1.5	12
2:00	1.5	2.4	2.4	1.8	1.8	1.5	1.5	13
3:00	1.5	2.4	2.4	1.8	1.7	1.5	1.5	13
4:00	1.5	2.4	2.4	1.9	1.7	1.5	1.5	13
5:00	1.5	2.4	2.5	1.8	1.8	1.5	1.5	13
6:00	2.9	3.9	3.8	3.2	2.2	1.5	1.5	19
7:00	4.7	6.1	6.0	5.3	4.3	1.5	1.5	29
8:00	7.1	8.6	8.2	8.2	5.9	1.6	1.5	41
9:00	8.0	9.6	8.7	8.9	6.6	1.6	1.5	45
10:00	8.2	9.6	8.7	9.1	6.8	1.6	1.5	46
11:00	8.2	9.9	8.7	9.4	7.1	1.6	1.5	46
12:00	8.2	10.2	8.6	9.6	6.9	1.6	1.5	47
13:00	8.6	11.3	8.7	10.1	6.8	1.6	1.5	49
14:00	8.4	11.4	8.4	9.7	6.9	1.7	1.5	48
15:00	7.9	10.7	7.8	9.0	6.4	1.5	1.5	45
16:00	6.0	8.7	6.1	6.9	4.5	1.5	1.5	35
17:00	3.6	5.2	3.8	3.9	2.7	1.5	1.5	22
18:00	2.4	3.5	2.5	2.8	2.0	1.5	1.5	16
19:00	2.0	2.9	2.1	2.0	1.6	1.6	1.5	14
20:00	1.9	2.7	2.0	1.9	1.5	1.6	1.5	13
21:00	1.9	2.5	1.9	1.9	1.5	1.5	1.5	13
22:00	1.9	2.4	1.9	1.8	1.5	1.5	1.5	12
23:00	1.9	2.4	1.8	1.8	1.5	1.5	1.5	12
Dagtotaal (kWh)	103	135	114	117	87	37	36	629

Figure 32: Hourly energy use per day during an average week of an office in a medical centre

(Plugwise, ErasmusMC 2012 from streamer D1.3).

Rule conversion from human to machine readable and computable

The design rules, once formulated, must be translated to meet two requirements. First, from an informal natural language the rules need to be converted to be understood by the application, the rules need to be machine readable. Second, the design rules need to be converted from functional relations to geometrical constraints, as discussed in the next paragraph 4.1.5.

For the conversion from natural language, the Streamer project uses a specific formal language, referred to as Domain-Specific Language (DSL), see Figure 33. This language is similar to rule formulation in structured natural language, it is tailored only to formulating Design Rules.
The DSL is then parsed to XML, as input for the Early Design Configurator, according to predefined rule types, see Figure 34. A tool called Knowledge Editor was developed in the Java Eclipse package, according to the following specification:

"The rule editor contains a semantic analyser that is responsible for converting the inputs text rules, which are processed by the lexical and syntactic analyser, into XML (Extensible Markup Language … ANTLR (Another Tool For Language Recognition … was used to implement the DSL and to create the parser that converts the rules into a machine-readable file; i.e., rules written using our DSL are parsed, and an XML file is created." (Sleiman2017NZEB)

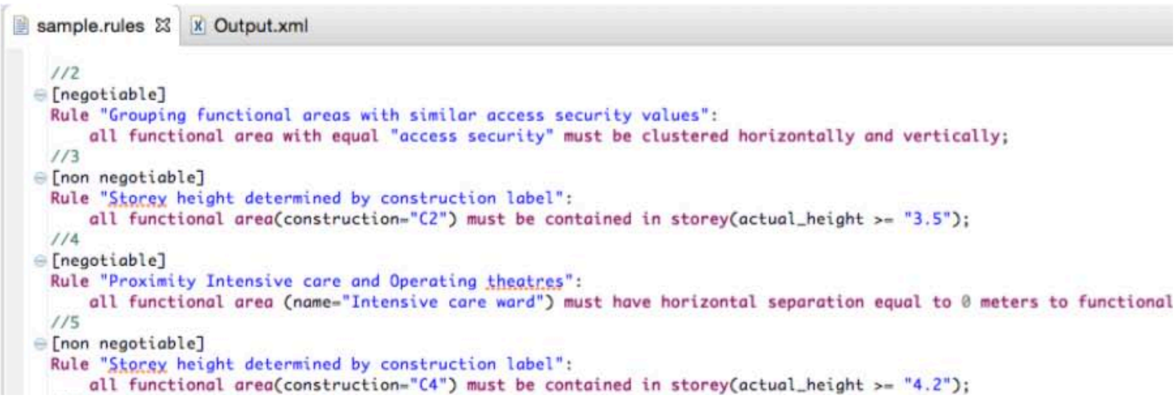

Figure 33: Sample design rules written in DSL, screen capture from Knowledge editor

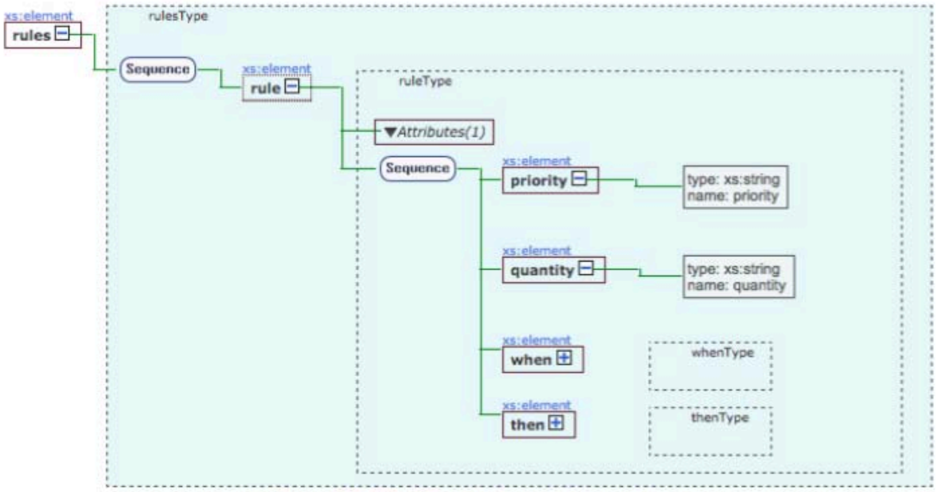

Figure 34: Definition of element rule types in XSD (Hempel & Sleiman)

An alternative method of creating design rules in XML uses an online google doc spreadsheet, which allows the user to pick from lists of predefined object types (Space, Storey, Wall) and values. (Werenstein, Streamer D5.6)

Other languages specifically suitable for reasoning are discussed in paragraph 3.2.3.

4.1.5 Configuring Early Designs

The next step in the Streamer workflow uses the Early Design Configurator, an experimental software application, to generate a series of building designs as simple IFC models, within the user defined constraints. A summary of the EDC workflow with in- and outputs is illustrated in Figure 35.

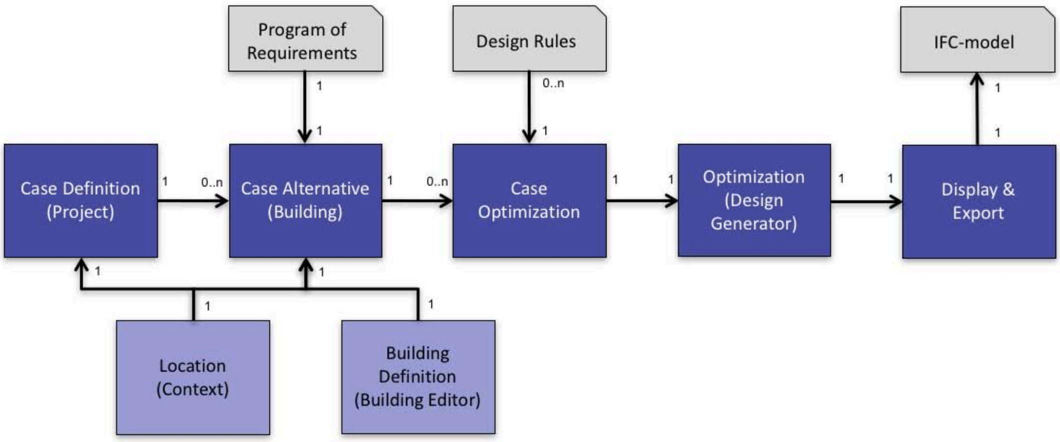

Figure 35: Description of Early Design Configurator, from Streamer review 2017

The process starts by creating and naming the project. The project location is located by a text search on [Openstreetmap](), which gives the user a direct visual feedback to create a geographical reference. On the site, for each project, several alternative cases (buildings) can be generated. To sufficiently define the projects constraints, a combination of semantic data and expert user input is required, see Table 11. Prepared, structured text files are uploaded and manual settings and constraints need to be defined graphically in the interface of the EDC application.

The semantic data consists of the following. The brief is defined and formalised both by the PoR table of room requirements (paragraph 4.1.3) and the design rules which may refer to room relations or also normative requirements (paragraph 4.1.4). In this phase, it is not possible to choose MEP systems, but based on the semantic labels, certain systems can be excluded as incompatible with certain space requirements (e.g. ComfortClass & HygieneClass). These exclusions are defined in a MEP system compatibility filter file.

The expert user then manually applies several constraints in the graphical user interface (GUI): The composition of the building shape in its articulations, each with an orientation, depth and height; the position of the street level access; the preferred building grid; Preferred articulation of corridor and rooms in the depth (single corridor; double loaded, double corridor; double loaded). Further constraints, especially relevant for renovation projects, may be vertical elements e.g. stairs, elevators, blocked spaces, immovable spaces.

Additionally, the EDC uses a resources library, containing object types and materials, to assign default wall, floor and roof construction types.

Table 11: Required input for the EDC

Input type	Description	Project/case
Preloaded	Libraries of default object types and materials	project
Structured CSV	PoR: Room requirements with semantic labels	project
Structured XML	Design rules	case
Structured CSV	MEP system compatibility filter	project
Manual URL	Project location, adjacencies	project
Manual GUI	Building shape, orientation, depth & height, distribution typology	case

The Evolutionary Algorithm

After the input has been completed, the process of layout generation can be initiated. The layouts are generated by an evolutionary algorithm. For each boundary (e.g. a building wing), a random layout is initiated. The algorithm runs in iterations.

Three operations are possible as mutations (Hempel2016):

- *The layout is randomly filled with rooms – either a room is added or removed.*
- *Rooms are resized randomly.*
- *The layout is changed to another random layout (KIT's version).*

The resulting floor plan is rated by calculating the satisfaction from all constraints, which is then compared to previous floor plans.

“The algorithm runs iteratively, where in each iteration, a room is randomly moved to a new position in the layout. If the resulting layout has a lower rating than the layout of the previous iteration, then the change is undone.” (Sleiman2017NZEB)

hard or soft constraints

To satisfy the PoR and the design rules in a sensible way, the EDC uses both soft, hard and combined constraints. Those design rules that if not respected would invalidate the design are classified as hard constraints. Those that are sufficiently relevant to be formulated as a rule, but not important enough to invalidate the design are considered soft constraints (Sleiman2017):

“A hard constraint is a Boolean condition that can be true or false; i.e., the violation of such a constraint results in an unacceptable value, which may be either one or even a higher value for cases where a layout should be discarded since the constraint is violated. An example of this kind of constraints is:

- *Space A must be within 20 m of Space B.*

A soft constraint results in a value that gets increasingly worse the more the constraint is violated. This kind of constraint requires a border value to normalize the output value, which in most cases is the maximum value of the input value. In special cases, where the border value is not the maximum value, the satisfaction may be above one. Examples for soft constraints are:

- *Space A needs to be as close as possible to Space B*
- *The walking distance between Space A and Space B must be as short as possible.*

A combined constraint combines both previous constraint calculation methods such that either a soft constraint is used inside the range of the border value or the result of the constraint is a bad value. Here are some examples cases:

- *Space A needs to be at least within 10 m of Space B.*
- *There need to be at least five spaces of Type C within N meters from Space A.”*

Scoring

Generated layouts that meet the design rules are assigned a higher score based on the priority of the rule, consequently they are maintained in the computer memory as less optimal solutions are dropped (Hempel 2015, Hempel 2016).

“This score is based on the actual placement of the rooms from the PoR, the size specified in the PoR, the fulfilment of the design rules, the connection of corridors and if the rooms have the right width/depth ratio of 60:40.” (Streamer,D1.6)

To attach an appropriate priority to each design rule is crucial for the design outcome, as the scored weight attached to the different design rules will determine the total score for each generated layout. As the EDC relies on evolutionary design computation and low scoring layouts will be dropped from memory.

Accept design or reiterate

"The evolutionary algorithm is run until it is interrupted either by the user or by meeting the termination criteria (maximum number of iterations). The next step is to either manually edit the layout or to change the priorities of the constraints and then continue to run the optimization algorithm. This can be repeated until the designer considers that one of the created designs is a good solution. However, the designer is also able to choose to continue the development of a given layout in two different directions, by cloning the layout and working with the copy. The copy can again be manually edited or its design rule priorities can be changed again."

"As evolutionary algorithm, the resulting layout may never be the best possible layout. Also, there exists no defined termination state, so the algorithm may run either for a defined time or a defined number of iterations. However, the current implementation allows the calculation of the global satisfaction to be extended by defining new constraints which integrate additional requirements."
(Hempel2015)

How do alternatives differ?

Differences between the alternatives will in part be the result of the users' design choices, either by changing the building volume for each case alternative, or within the same volume, by applying different sets of design rules for each case. For instance, the same set of rules can be applied while the priority for each rule is changed. For another part differences are due to the heuristic optimization process of the evolutionary algorithm which packs the spaces within the given boundaries, starting out randomly for each alternative.

No cross breeding

The specific algorithm of the EDC differs from the general description Cross breeding designs like in chapter 3.4 would not make sense. Because the parameters or "genes" are not clearly distinguishable. Each design alternative consists of a complete set of rooms which can be only self-referential. Each alternative design is being improved as the computer sees alternatives, but some designs may get stuck in some local optimum. Therefore: *"The EDC works on several layouts in parallel, and the worst layout is regularly reset completely and restarted from scratch."* (Sleiman2017NZEB)

From relationship to computable design rule

The second translation of the design rules, is the conversion from functional, relational rules as an architect would use to computable geometrical requirements that can influence the spatial placement in the model of rooms or groups of rooms. (Hempel2016)

In most cases, a translation is required from ambiguous architectural convention to unambiguous computable rule. For example, the term clustering for architects is may be clear enough to indicate grouping spaces with similar functions. When translated into computation it turns out to be ambiguous. Does it mean spaces should share adjacent walls, or can they be on opposite sides of the same corridor?

"Generally speaking, the rules can be divided into two categories: Rules with unary and rules with binary relations. The unary relations relate a group of rooms to itself; binary relations relate one group of rooms to another group."(Hempel2016)

Binary relations

- Must be on same/different storey as
- Must be contained in storey
- Must be directly above/below
- Must be partly above/below
- Must have travelling distance
- Must have separation of

Unary relations

- Must be clustered horizontally
- Must be clustered horizontally or vertically
- Must have separation to outer boundary of

Informal rules are defined in Streamer D5.5

Example rule translation

The interpretation and transformation of relational design rules into calculable geometrical rules is shown below. The walking distance between to spaces is measured from the centre to the centre of these rooms, through the middle of the corridors (Figure 36, left). The maximum separation between two rooms of different functional area, identified by the colours, is measured between the most distant rooms of these functional areas (Figure 36, right).

Figure 36: Walking distance (left) and Max. separation (right), from Hempel 2016

4.1.6 Checking the designs against semantic rules

"… 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." (Streamer D5.5)

Advantages of evolutionary design algorithms include their ability to deal with *over constrained* problems. This is likely to be useful within the hospital context described above. The algorithm deals with an over constrained design problem by also generating designs that do not meet all the constraints. During the generation of design alternatives, some designs may be generated that do not meet all the requirements or satisfy all the design rules. Therefore, we will have to validate the design to check which rules are satisfied.

In later design phases changes may be made without a complete awareness of all the requirements. A design validator to establish whether all the requirements of the brief, and all the building codes are still respected.

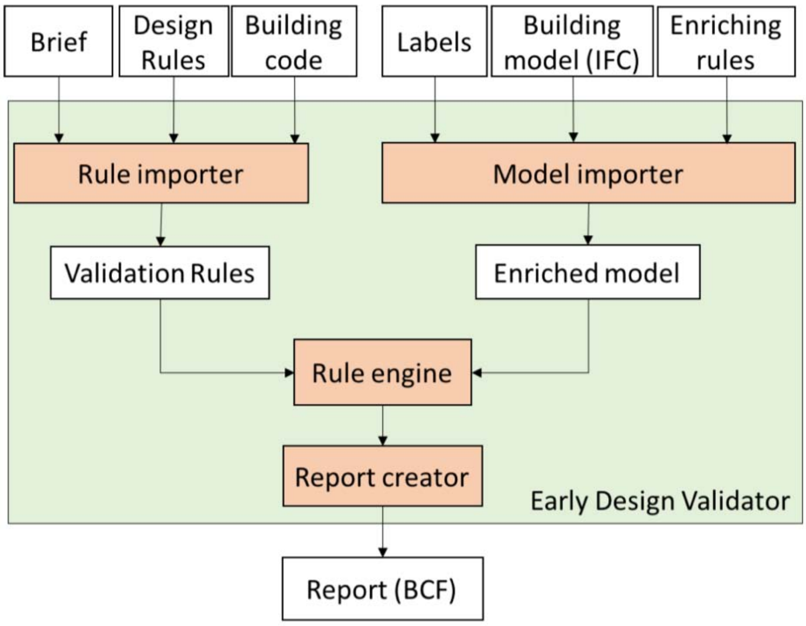

Figure 37: Design validator system architecture, Sleiman2017

Rule checking with re-usable and adjustable rules is a current research area as noted in paragraph 3.2.3. Design rules may used to generate a design, checking rules to validate the application of design rules and the application of building code (Eastman2015). Enriching rules may be used to add or infer information based on other information that is already present. The new ifcOWL standard is specifically capable of this type of logic so implementations are expected in the near future (Pauwels2016).

In the streamer process a reasoning rule engine, the Early Design Validator was introduced to detect whether all the spaces present in the model meet the PoR requirements (count, area, labels), which if any design rules are not satisfied and if any building codes are not met. Regarding the building codes only a small sample has been implemented based on the information that is available in an EDC generated file (Sleiman2017). The EDV prototype also allows user defined rules. More detailed building codes and further BIM delivery requirements have been defined in other research (Hjelseth2010, Zhang2014, Zhang2015, Roxin2016).

The EDV follows a four-stage process as suggested by Eastman et al (Sleiman2017). In the first stage the PoR text file and the design rule XML and eventual building codes are converted into an object-oriented language by a rule importer module.

In the second stage the model is turned into a graph model (see Figure 38) and enriched by calculation and inference. Calculations can derive quantities such as the volume based on values present in the file. Inference is a means of reasoning through the application of logic. Enrichment by inference could mean for example using the semantic labels, inferring more information than is contained in the model. For example, for rooms with a high hygienic class will require specific finishes that can be cleaned, and certain MEP systems e.g. radiators may be excluded as being too difficult to clean. These reasoning rules create explicit new information. In this stage, a semantic label may be enriched with all the possible properties that can be derived.

In the third stage the reasoning engine applies the rules to the graph model to verify which rules pass validation. If sufficient information for a certain rule is not available that rule is skipped.

In the fourth and final stage the failed checks are collected and automatically reported as a collection of Bim Collaboration Files (BCF).

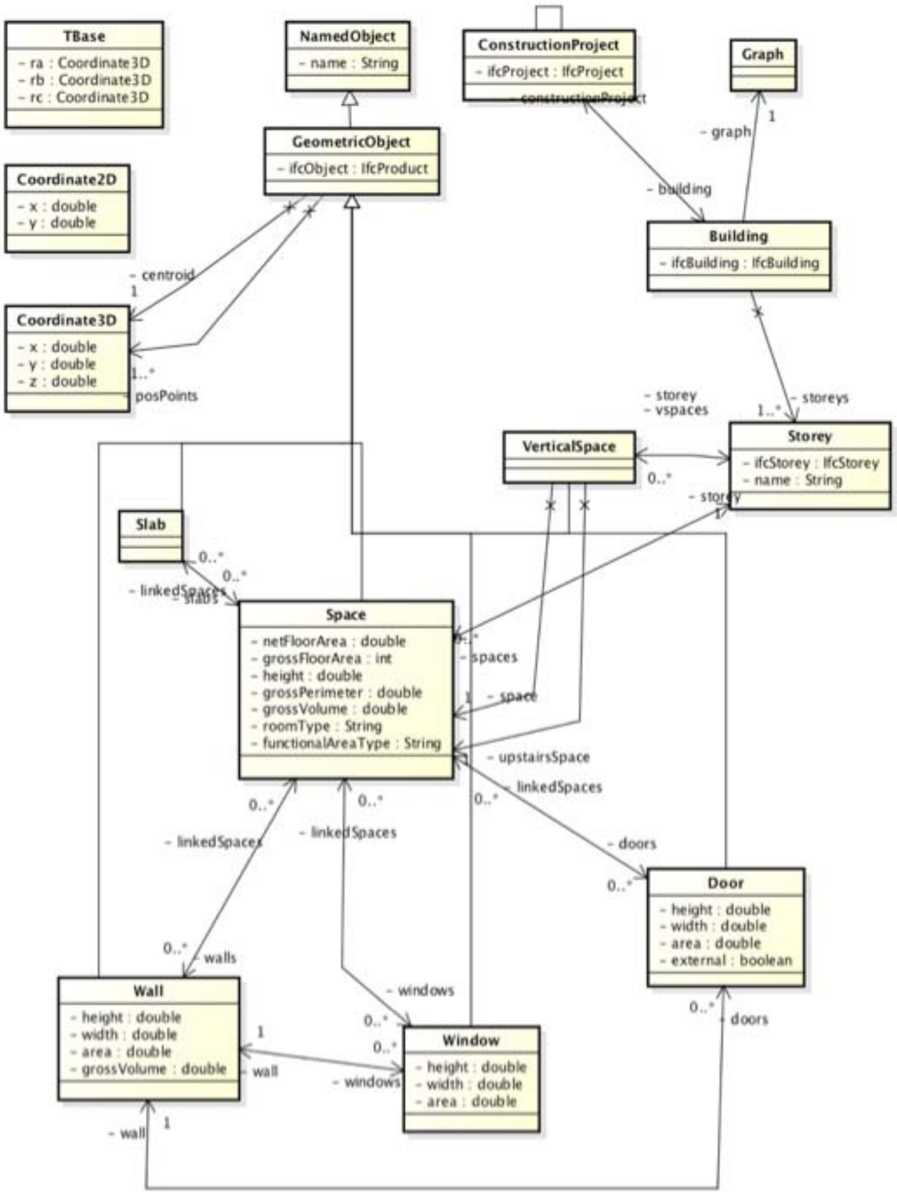

Figure 38: Internal model of information in EDV (Sleiman 2017 .pptx)

4.1.7 Selecting MEP systems & Energy efficient Building solutions

To compare different technological solutions for the same building layout, the architect and climate consultant experts may choose to assign different solutions than the default assigned by the EDC. It may be useful to differentiate solutions based on the location within the building or on the orientation. Within the EDC generated layout, a degree of top-down uniformity may be desired, e.g. increasing uniformity between the windows in the same façade, or between MEP systems on the same floor. In a similar way, room labels in the same Functional Area can be realigned i.e. changed from the default value.

The two component types, MEP systems and Energy efficient Building (EeB) solutions are treated in a slightly different way. The suitable MEP systems can be determined by looking at the MEP system compatibility contained as a room label in the IFC file. Selected rooms that the expert decides will share the same system may be grouped in an IFC zone. The IFC specification of a zone allows the same room to be part of different zones at the same time. A viewer such as eveBim or SimpleBIM can be used to create zones that are intended to share the same MEP systems.

The building components (walls, windows, floors, roof) that are present in the IFC file can be all selected for the entire project, or partly based on their labels and/or location. To these components different physical and cost properties can be assigned together with an EeB code, see Table 12.

Table 12: Extract of EeB solutions table, attachment to Streamer D2.6

EeB System Code	Ifc Element	Decription	Thermal Transmittance W/m²·K IfcThermalTransmittance	Glazing Area Fraction (%) % IfcGlazingAreaFraction	Specific Heat capacity J/m²·K IfcSpecificHeatCapacity
EeB_wl_01	IfcWallStandardCase	Outside Wall minimum.	0,80	N/A	330000
EeB_wl_02	IfcWallStandardCase	Outside Wall good.	0,20	N/A	350000
EeB_wl_03	IfcWallStandardCase	Outside Wall excellent.	0,10	N/A	370000
EeB_gf_01	IfcSlabStandardCase	Baseslab minimum.	1,00	N/A	450000
EeB_gf_02	IfcSlabStandardCase	Baseslab good.	0,25	N/A	460000
EeB_gf_03	IfcSlabStandardCase	Baseslab excellent.	0,10	N/A	460000
EeB_rf_01	IfcSlabStandardCase	Roof minimum.	0,76	N/A	400000
EeB_rf_02	IfcSlabStandardCase	Roof good.	0,17	N/A	430000
EeB_rf_03	IfcSlabStandardCase	Roof excellent.	0,1	N/A	450000
EeB_wi_01	IfcWindowStandCase	Window minimum.	1,60	100	34000
EeB_wi_02	IfcWindowStandCase	Window good.	1,10	100	34000
EeB_wi_03	IfcWindowStandCase	Window excellent.	0,70	100	37000

This method increases the levels human interaction and allows considerations of increasing detail based on the same generated layouts. First, the output of the EDC can be directly simulated using default values to create a baseline energy consumption. This is entirely automated in the TECT tool (see paragraph 4.1.9). In the next steps experts can try different configurations of MEP systems and EeB solutions for the same spatial layout configuration.

The changes to the IFC model generated by the EDC can be assigned manually by the architect or climate consultant in an IFC editor such as eveBIM or SimpleBim.

4.1.8 Comparing design alternatives on Key Performance Indicators (KPI's)

An automated semantic design process as described above is capable of generating many designs, limited only by computational time. The designs that are generated in the EDC based on the design rules will meet the functional criteria in part or in full. The design validator helps eliminating the designs that do not meet basic standards. Then, what criteria will distinguish the generated designs?

Beyond the strictly functional, an evaluation of other criteria is needed. Three categories of performance have been identified in the Streamer project: Energy performance, Project finance, quality. Each category is synthesized in a single numeric Key Performance Indicator, calculated from multiple Performance Indicators (Schreuder2015).

Three criteria have been used for the selection of the Performance Indicators: Consistency of unit of measurement in design and operation phases, calculation preferably with existing methods, the calculation should be based on data available in the IFC file.

The selected Key Performance Indicators, and underlying Performance Indicators in the Streamer project are:

Energy performance:

1. Energy efficiency
2. Carbon emission efficiency

Financial performance:

3. Life cycle costs

Quality performance:

4. Patient satisfaction
5. Overall quality
6. Thermal comfort
7. Operational efficiency

A balanced weighing of criteria is crucial to obtaining balanced results, as remarked in chapter 3.6. This will avoid results skewed to optimize single aspects.

It was investigated whether a direct link between the label requirement and the selected Performance Indicators could be found. The response is negative, as the chosen design solution lies between the label and the performance indicators. Only when there is a design choice can the performance indicator be found. Especially the quality performance indicators have been elusive as there are no models to predict patient satisfaction or overall quality. This is more true in the early design phase when many detailed choices that influence perceived quality still need to be taken.

The next paragraphs will explain briefly how these energy and financial performance indicators have been determined.

4.1.9 KPI: Evaluating energy performance

Conventional design processes use dynamic, year-round, whole building energy simulations to predict comfort levels, energy demand, energy consumption, etc. As these simulations are very time consuming, they require an expert to manually insert many of the settings. Thus, they will be performed later in the process, typically in the design development phase. Anticipating this information regarding energy performance, allowing it to influence the early phase design iterations is one of the objectives of the proposed Semantic BIM workflow.

To allow the evaluation of many alternatives an automated simulation tool was used. The TNO Energy Calculation Tool (TECT) is being developed to calculate performance of IFC models according to the current European standard: Energy Performance of Buildings (EN ISO 52016-1 and EN ISO 52010-1). It performs hourly calculations for each room in the project.

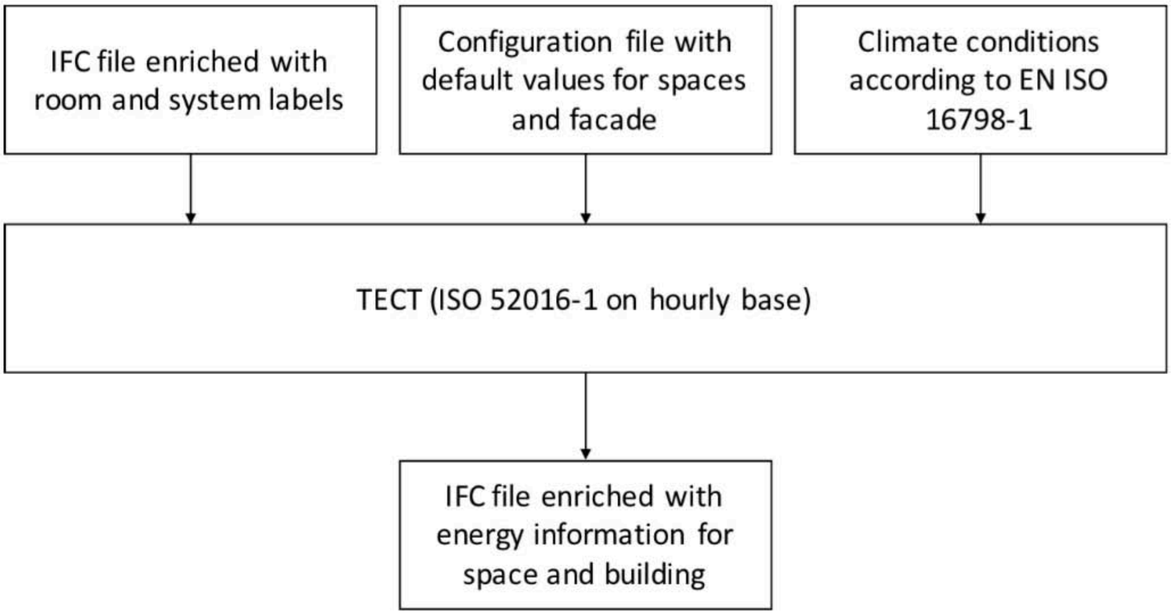

Figure 39: Input and output of TECT, from Sleiman et al. 2017

The TECT tool was programmed to interpret the information flow of the Streamer project. It loads the IFC model, interprets the semantic labels of rooms and MEP systems to configure the simulation. It interprets location and orientation and will load the correct weather file automatically from the geographical position embedded in the IFC file. The TECT tool interprets the geometry and the first level space boundaries supplied by the EDC.

The used version also checks for Streamer label values and where label values are not found, default values are used from an external configuration file. This is useful in the case of spaces not requested in the PoR such as corridors which may be without semantic labels. The default values can also be applied to IFC models without any labels, but of course in that case TECT doesn't use any detailed information at the room level. Streamer labels contain a property set for each space with information about MEP systems: heat generation; heat emission; cold generation; cold emission, and EeB solutions, specifying ventilation and the façade.

The TECT tool analysis results (and any default room label values it adds), are written back to the IFC file for downstream use. The simulation outcome contains heat demand, cold demand, energy consumption by cooling system and energy consumption by the heating system (all values in MJ/year). It also outputs Max Power Heat Demand and Max Power Cold Demand as properties.

Comparison is possible only when using the same simulation tools and settings on the alternative designs, a comparison of simulation tools show the results for the same input differ too much otherwise.

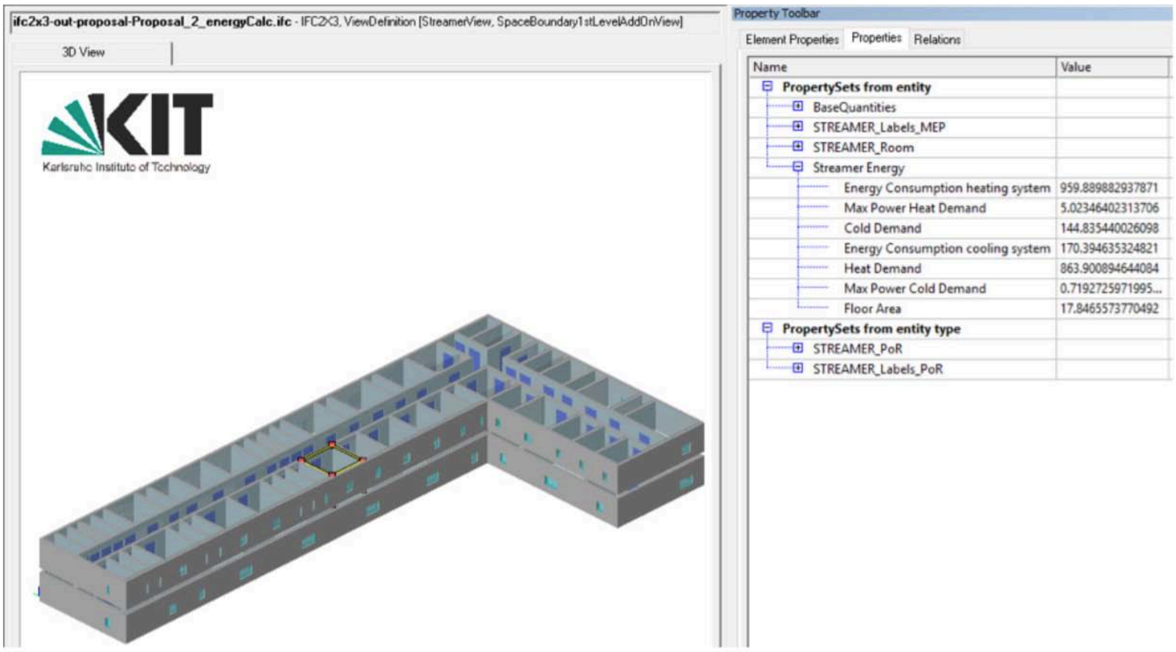

Figure 40:Energy simulation results stored for each space

Two step approach

The described method investigates the contribution of automated processes and information exchanges in the early design phase. Where the model exchange requires a large number of manual settings for each design alternative this process is interrupted. An example of a step requiring many manual settings is a full dynamic energy simulation.

As a solution, a two-step approach is introduced. In the first automated phase the energy simulation is not predictive but only comparative. In this phase, the simulation uses default settings, either based on room labels in the IFC or default values that are added automatically. This creates a baseline value for energy consumption.

In the second phase, some manual intervention by domain experts is possible to rationalize the MEP systems and EeB solutions, see paragraph 4.1.7. One or more options can be created. This file with partially modified labels, MEP systems and EeB solutions is again suitable for an automated energy simulation. The energy performance of the manipulated options can be compared with the performance of the baseline model.

For one or more of the best performing models a conventional full dynamic energy simulation with all its manual settings can then be performed.

4.1.10 KPI: Life cycle costing

To compare design alternatives in a meaningful way, cost indicators need to be considered. The method used is Life Cycle Costing (LCC). This considers not just the initial investment cost, but attempts to gather all costs in the buildings life cycle, including: design, construction, operation, maintenance, demolition. These costs are foreseen on a yearly basis to allow estimating financial commitment.

Due to the difficulties of collecting this information for each of the countries involved, the Streamer implementation illustrates feasibility by using a simplified model with only investment and operational costs. The costs are derived from Dutch hospital buildings.

The costs used are, investment costs, referred to as CAPEX, the amount of invested money that needs to be financed. And operational costs, referred to as OPEX, which includes all the operation and maintenance costs.
.
Some other costs such as demolition or renovation, and residual value are not considered in the implementation, are hard to predict in this phase. Financing costs and revenue of exploitation are too dependent on the context.

The costs can be directly assigned in the Decision Support System based on some of the Streamer labels in the IFC file. Costs are assigned from BouwcollegeLayer, MEP systems and EeB solutions.

4.1.11 Comparing design alternatives with the decision support tool

Decision Support systems originally were developed in the form of spreadsheets. These systems have transformed into a web based interface, which can be visualised as a dashboard, see Figure 41. It retrieves selected data from a database and presents it graphically and synthetically, similar to a dashboard.

For the early stage building design, besides energy consumption, cost and quality aspects guarantee a balanced result. (Streamer Deliverable D4.1). Data for different indicators (Performance Indicators) can be aggregated, and multiplied with weighing factors. combined into Key Performance Indicators (KPI's). These may be again aggregated and multiplied with higher level weighing factors. A synthesis is found in the selection of a limited number of Key Performance Indicators (KPI's) which can be used to compare design alternatives.

In order to obtain balanced results competing factors should be considered, as proposed in "The balanced Scorecard" (KaplanNorton1996). A scorecard in which certain factors will be unreasonably neglected, will strengthen single aspects and thus lead to a false optimum.

Assigning the weighing factors and aggregation requires knowledge of how the underlying mechanisms influence the Performance Indicators, this requires expert knowledge. The decision maker using the interface should understand the way the data is being handled to avoid the risks of oversimplification.

The Streamer dashboard uses a web application named RE suite. The dials on display can be adjusted to the needs of the user, see Figure 41. It also offers an IFC viewer allowing the user to filter spaces based on their semantic labels.

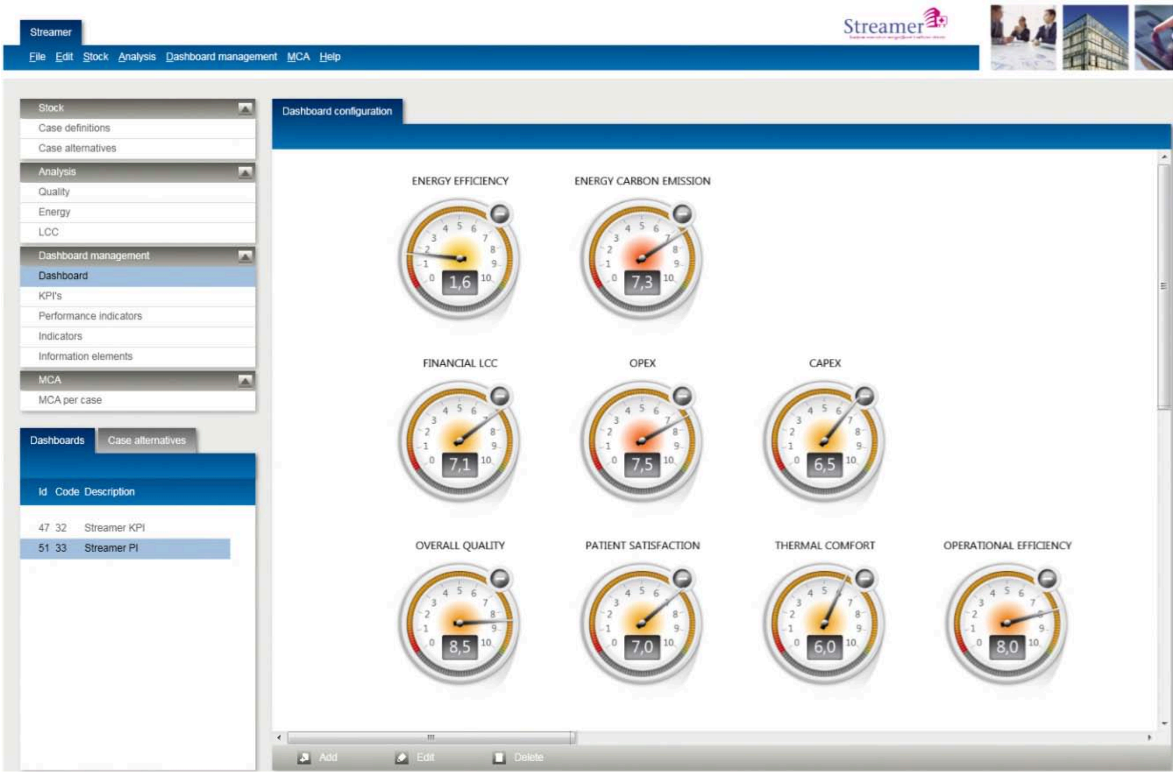

Figure 41: Illustration of KPI Dashboard in the DST, from Streamer D3.6

In this process, in the Early Design Phase, hard data has been generated to compare design alternatives on energy and financial performance indicators. The quality criteria have been excluded due to the lack of information that belongs to the phase. It will be up to the architect to evaluate, and balance their potential.

If an early building design is found to meet the expected criteria it can be selected to be developed in the next design phase. If, on the other hand, none of the generated designs satisfy the criteria it may trigger another early design cycle. The drive to obtain a more satisfactory design may even lead to reconsider the initial input, such as the brief or the design rules. The fact that this information is explicitly available should aid a review.

4.2 Retrofitting case of San Luca pavilion, Careggi hospital

During a buildings' operational time its performance decreases, while organisational needs change. As the gap between the building quality and occupant expectation grows, different scenarios may be considered. These scenarios may include moving to a new or existing building, or improving the same building to meet different and higher standards. The scenarios that refer to adjusting an existing building are referred to as retrofitting.

The components that make up a building tend to become obsolete periodically. The problem is that these periods do not coincide among them, as Brand noted in *How Buildings Learn: What happens after they're built (1995).* He refers to components as belonging to layers: Site; Structure; Skin; Services; Space Plan; Stuff. Brand studies why some buildings are easier and less costly to adapt than others, proposing strategies to deal with change over time. Flexibility and adaptability of a building are defined by those components that are fixed or that are hard to change, as follows by the analysis of Leupen in *Kader en generieke ruimte (2002)*.

There are several reasons for the increased occurrence of retrofitting. Within certain parameters, adjusting an existing building may come at a lower financial and ecological cost than building a new one. To reduce CO2 emissions, it would be more effective to improve the energy performance of many existing buildings than to only build better new buildings. The hospital district is no exception.

Over time, modern buildings from the second half of the 20th century are increasingly found to be lacking in performance. Performance of building facades and MEP services has greatly improved over the last decades, achieving higher levels of comfort. Instead the building structure may be found to be still adequate if it was designed with some extra capacity. Typically, modern concrete buildings are suitable for more than one use, as they have a regular frame structure which places fewer constraints than older buildings with internal bearing walls.

The Streamer project illustrated in the previous chapter develops a toolbox of semantic design instruments: standardised room and functional area types; semantic labels, design rules, automated layout generation, design validation tools and a decision support tool. How can these tools be applied in the context of energy renovation?

The starting point for any choice on should be an analysis of the fitness of the current layout, the energy performance of the building and the running costs. In the case of the renovation of existing buildings, these same tools may be applied also partially. The labelling method can be applied also to evaluate an existing layout as explained in the following paragraphs till 4.2.4. The case study of the San Luca pavilion will serve to illustrate.

The evolutionary algorithm can be used to generate alternative layouts as explained in paragraph 4.2.5. This may be within a series of constraints determined by the existing building, possibly modified by interventions on the volume, the floors or the façade.

4.2.1 Mapping the existing spaces

As a first step, a survey and detailed model of the existing San Luca Vecchio building were prepared in Archicad. The level of detail exceeded the level required, so for the Streamer process the model needed to be simplified. To resolve the encountered problems during export, several steps were undertaken. The door and window objects were substituted with Archicad IFC library objects, the amount of wall types has been reduced and the internal/external attribute has been corrected.

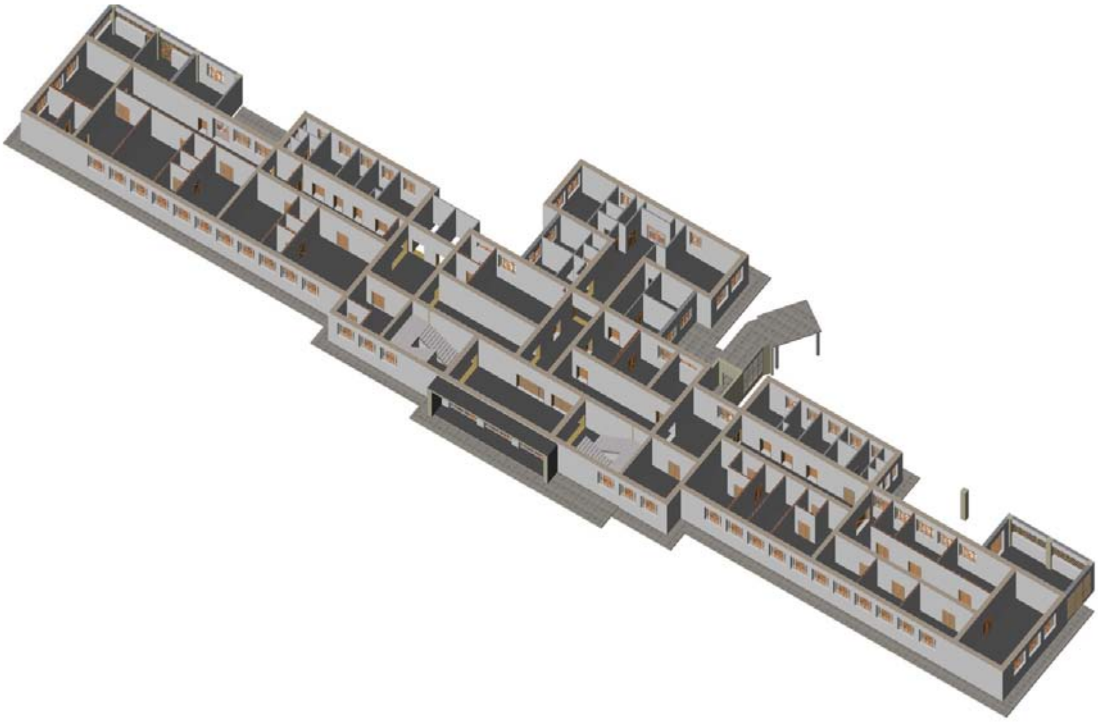

Figure 42: IFC model of the San Luca Vecchio, Careggi

Energy simulation tools importing IFC Simergy Pro and IDA ICE were trialed but not used, due to import problems. To establish a baseline energy performance, a full dynamic energy simulation was performed in Designbuilder. To prepare the model an IFC was imported into Revit, where second level space boundaries were added, the export from Revit to Designbuilder was then made in gbXML (a format that maintains geometry but not room level information).

The simplification of the IFC file, verification of its suitability as an energy model and data enrichment were performed with SimpleBim. This has proven to be a very flexible tool which allowed to eliminate all the geometry and information not required for the successive steps. SimpleBim was used to validate aspects of the IFC schema. It also offered some tools to check whether the model was ready for energy simulation, for example whether all the components had a type and whether the building envelope was closed.

In addition to manual interventions through the graphic user interface (GUI), SimpleBim provides an excel template to structure input. The application then intervenes on the IFC file with this input. In this accessible way, it is possible to:

- Control IFC export settings;
- Set the model author for each modified object;
- Set up the Model View, defining which objects the file should contain;
- Validate IFC properties by checking for allowed values;
- Enrich the file, i.e. add properties to objects based on other properties;
- Substitute values;
- Group objects based on their name or properties; according to the Ifc schema the group can then be mapped to different classes, e.g. IfcZone, IfcGroup, IfcAsset, IfcBuildingSystem, IfcDistributionSystem.

After being simplified and validated, the model was enriched with Streamer data in two steps. First, Streamer labels were attached to all the rooms in the model based on the room number, see Figure 43. The Streamer RoomType as mapped from SDU class was used to select the labels, see paragraph 4.2.2.

Object Or Group [+]	Space						
Property Name or Key	Space Number	Room Type	FunctionalArea Type	Amount	Bouwcollege Layer	Hygienic Class	Access Security
Operator	Match = equals	Set	Set	Set	Set	Set	Set
	16_A_001	ConferenceRoom	ConferenceRoom	1	O	H2	A2
	16_A_001a	Corridor	Corridor	1	H	H1	A1
	16_A_001b	Toilet	Toilet	1	I	H4	A2
	16_A_001c	ToiletDisabledPeople	ToiletDisabledPeople	1	I	H4	A2

Figure 43: attach labels to rooms with SimpleBim

All the rooms in the model of the existing building were thus provided with Streamer labels, which are meaningful however only to tools adapted to the Streamer workflow. To increase the useable data in the IFC file the Streamer labels were then used to define standard property sets. Precise indications at the time of the file manipulation were found in the Streamer deliverable D1.6, since updated in the google sheet "D5.6 Framework for the open-source library of parametric design solutions". Each abstract Streamer label value implies other real requirements, for the given example of the ComfortClass these define: relation to the building envelope, mechanical ventilation rate, min. and max. temperature setpoints, light levels, relative humidity levels, acceptable indoor noise levels.
For each label value, different physical requirements were added to the file, see Figure 44: property enrichment based on label values with SimpleBim.

Object Or Group [+]	Space				
Property Name or Key	**Comfort class**	**Space Temperature Min**	**Space Temperature Max**	**LightingRequirement**	**Mechanical VentilationRate**
Operator	Match = equals	Set	Set	Set	Set
	CT1	<no value>	<no value>	NOTDEFINED	<no value>
	CT2	<no value>	<no value>	DIRECT DAYLIGHT	<no value>
	CT3	20	<no value>	DIRECT DAYLIGHT	10
	CT4	20	24	DIRECT DAYLIGHT	10
	CT5	20	24	DIRECT DAYLIGHT	10
	CT6	18	24	NOTDEFINED	18
	CT7	18	24	NOTDEFINED	60
	CT8	<no value>	<no value>	NOTDEFINED	<no value>

Figure 44: property enrichment based on label values with SimpleBim

4.2.2 Labelling space requirements

The *San Lucca Vecchio* building, distributed over three storeys, contains a total of 266 spaces. These belong to 39 space classes according to the *Sub Destinazione d'Uso* (SDU) classification used by Careggi. Within the Streamer context for a total of 88 Streamer room types have been classified, for each room type the space requirements have been defined. So, it is by mapping the Italian space class to the Streamer room type that the semantic label requirements can be assigned. These space requirements express the spatial typology, environmental conditions and building quality the room *should* offer, not necessarily the quality that will be found in an existing building, in this case of the *San Lucca Vecchio,* built in the 1960's.

Table 13: Mapping use classes to Streamer room labels as per room type

SDU CLASS	STREAMERROOMTYPE	AMOUNT	STREAMER LABELS
Accettazione	Reception	2	Office, H4, A1, U2, EQ1, C1, CT3,
Altro	PatientRoomIntensiveCare	13	Hotel, H2, A2, U4, EQ6, C1, CT4
Altro	Laboratory	16	HotFloor, H5, A5, U3, EQ6, C4, CT6
Altro	Toilet	2	Industry, H4, A2 , U4, EQ1, C1, CT6
Altro	StoreRoom	13	Industry, H2, A5 , U1, EQ1, C1, CT6
Altro	TechnicalRoom	4	Industry, H1, A5 , U3, EQ1, C4, CT6
Ambulatorio Visita	ConsultationExaminationRoom	3	Office, H3, A2 , U1, EQ1, C1, CT3
Antibagno	AnteRoom	2	Hotel, H1, A2 , U4, EQ1, C1, CT4
Archivio Cartaceo	Archives	1	Office, H1, A5 , U1, EQ1, C1, CT3
Attesa Parenti	WaitingRoom	1	Office, H1, A2 , U1, EQ1, C1, CT3
Attesa Pazienti	WaitingRoom	1	Office, H1, A2 , U1, EQ1, C1, CT3
Aula	GroupRoom	1	Office, H2, A2 , U1, EQ1, C1, CT3
Connettivo Orizzontale	Corridor	42	Hotel, H1, A1 , U4, EQ1, C1, CT2
Connettivo Verticale	Stairs	6	Hotel, H1, A1 , U4, EQ1, C1, CT2
Coordinatore Infermieristico	NursingStation	4	Hotel, H2, A5 , U4, EQ1, C1, CT4
Degenza con WC	PatientRoom	15	Hotel, H2, A2 , U4, EQ2, C1, CT4

Doccia	Toilet	1	Industry, H4, A2 , U4, EQ1, C1, CT6
Ecografia	ConsultationExaminationRoom	5	Office, H3, A2 , U1, EQ1, C1, CT3
Locale Caldaia / UTA	TechnicalRoom	3	Industry, H1, A5 , U3, EQ1, C4, CT6
Locale Infermieri	NursingStation	6	Hotel, H2, A5 , U4, EQ1, C1, CT4
Locale Medici/Refertazione	NursingStation	1	Hotel, H2, A5 , U4, EQ1, C1, CT4
Locali Personale Sanitario	Office	1	Office, H2, A4 , U1, EQ1, C1, CT3
Materiale Sporco	StoreRoom	3	Industry, H2, A5 , U1, EQ1, C1, CT6
Medicheria	Treatment room	4	HotFloor, H4, A3 , U4, EQ6, C1, CT4
Pulizie	StoreRoom	5	Industry, H2, A5 , U1, EQ1, C1, CT6
Sale Riunioni	ConferenceRoom	3	Office, H2, A2, U1, EQ1, C1, CT3
Segreteria	Office	1	Office , H2, A4 , U1, EQ1, C1, CT3
Servizi Igienici per il Personale	Toilet	21	Industry, H4, A2 , U4, EQ1, C1, CT6
Servizi per Pazienti	Toilet	17	Industry, H4, A2 , U4, EQ1, C1, CT6
Servizi per Pazienti (Disabili)	ToiletDisabledPeople	11	Industry, H4, A2 , U4, EQ1, C1, CT6
Servizi per Pazienti H	ToiletDisabledPeople	3	Industry, H4, A2 , U4, EQ1, C1, CT6
Servizi Pubblici	Toilet	7	Industry, H4, A2 , U4, EQ1, C1, CT6
Spogliatoio per il personale	ChangingRoomPersonnel	4	Industry, H1, A5 , U4, EQ1, C1, CT6
Spogliatoio per paziente	ChangingRoomPersonnel	3	Industry, H1, A5 , U4, EQ1, C1, CT6
Strumentario Chirurgico	SterileStore	1	Industry, H3, A5 , U3, EQ1, C1, CT6
Studio Medico	Office	18	Office, H2, A4 , U1, EQ1, C1, CT3
Terapia Intensiva	PatientRoomIntensiveCare	8	Hotel, H2, A2 , U4, EQ6, C1, CT4
Tisaneria	Kitchenette	6	Industry, H2, A4 , U1, EQ3, C1, CT6
Ufficio	Office	6	Office, H2, A4 , U1, EQ1, C1, CT3
Vuota	UtilityRoom	1	Industry, H4, A5 , U4, EQ3, C1, CT3
Zona Relax personale	RestingRoomPersonnel	1	Office, H2, A4 , U4, EQ1, C1, CT3

4.2.3 SACS© Careggi database

The large scale of hospitals can be managed only with trusted and current data. The Careggi hospital is the biggest employer in Florence. It must control over 15.000 rooms in 52 buildings. To manage all this, a database that combines aspects of facility management, hospital staff and equipment has already been foreseen.

SACS© is a software tool developed in house by the Careggi hospital in cooperation with the University of Florence since 2003. It is unconventional because it stores all information in DWG drawing files, rather than in a more common GIS format (Iadanza2009). Normally, drawings are only used by technical staff for the purpose of facility management. In this case, on top of architectural information, information is also added regarding medical equipment and staff organization through the SACS© interface.

To add a building to the database, the process follows the flowchart below, see Figure 45. Once the data has been collected in the DWG file, SACS© can be exported in different formats based on use: link it to other data in the Hospital Information System (HIS) database, publish reports and make HTML and DWF files. Since all the data is held in the

DWG file, the database can be rebuilt by copying the information from the DWG files. The use of single database simplifies procedures.

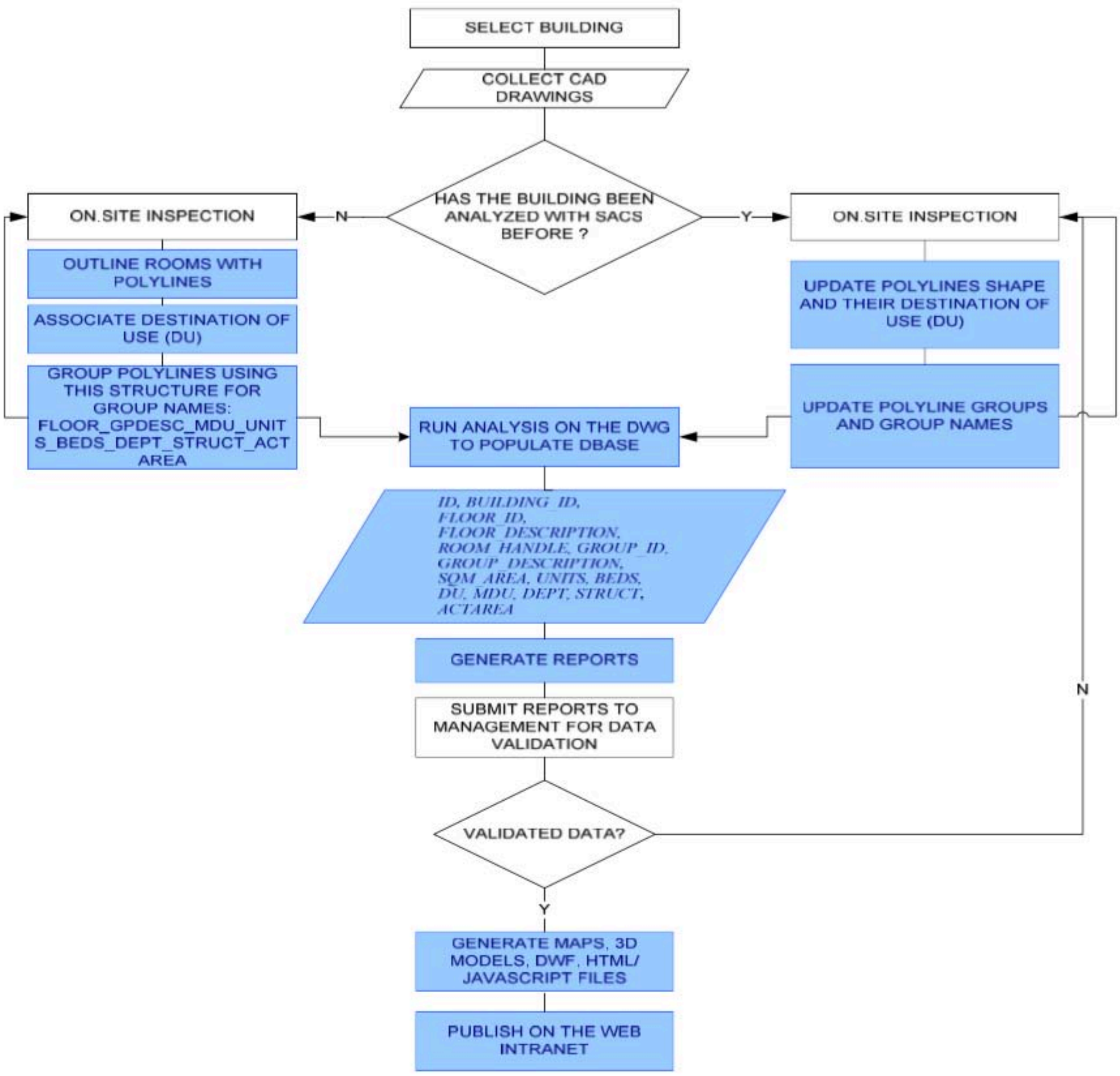

Figure 45: Process flowchart. (Iadanza2009unconventional)

Blue steps are performed using SACS©

DWG files offer the possibility of data storage as object attributes linked to graphical elements. This data is held by blocks or polylines. In the case of SACS© the group function is named as an array of field names, separated by underscores with the following syntax:

FLOOR_GPDESC_MDU_UNITS_BEDS_DEPT_STRUCT_ACTAREA

Where the fields can be described as follows:

- FLOOR is the level inside the building, useful for multi-floor drawings
- GPDESC is a brief description of the polylines group
- MDU is the prevalent destination of use
- UNITS is the actual number of rooms in the group
- BEDS is the total number of ward beds in the group
- DEPT is the department code

- STRUCT are codes representing the operative structures that use the rooms
- ACTAREA is a number code that represents the Activity Area.

The description in the Streamer deliverable 7.5 is more comprehensive and understandable: *"The software maps departments and relative Operative Space Units, purpose of use, healthcare technologies and environmental comforts, grouping info by single rooms and homogeneous areas, giving quantitative and qualitative results (such as surfaces, heights and volumes, Key Performance Indicators, etc.)... ...(spreading of operative units, activity areas and departments, beds, detailed destination of use and environmental comforts of each room)."*

Within the hospital this information contained in an SQL database server can be accessed through the intranet. It is therefore available to hospital management, healthcare professionals, facility management and technicians. It is accessed using EUREKA, a web-based search engine that allows for complex queries.

Recently an android app has been developed, Careggi Smart Hospital which opens to the general public some of the information from the database. For example, it assists wayfinding in the hospital with a map visualisation (Luschi2014).

In future, the transfer of SACS© from a DWG based to a BIM based process can be foreseen. New possibilities could be unlocked by using a BIM based process and viewer, which could facilitate other ways of viewing and handling the information.

In the Streamer case study of the San Luca hospital, the semantic labels of the room requirements and space properties have been added to the database to explore the possibilities for renovation with improvement of energy performance.

4.2.4 Comparing actual space properties vs. function requirements

The Semantic labels and room types offer a method of evaluating the suitability of an existing space layout. Analysing fitness by comparing room requirements derived from function to space properties. This step serves to evaluate how well the existing functional layout matches the properties of the individual spaces.

Once an analysis has established how far removed the existing layout is from the required performance, this is a measurement for both for understanding the suitability of the current layout, as well as a measure for the over- or underperformance of the building. This may also be visualised, see Figure 46. Visualising the discrepancy between space property and function requirement could also help to understand the shortcomings of the existing spaces.

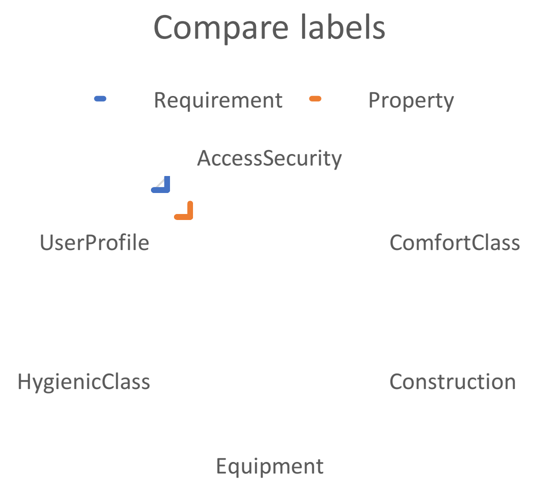

Figure 46: Compare function requirement to space property labels

Another way to use the labels is as a graphic aid to visualize clusters of labels. The UserProfile label shown in Figure 47, describes the active hours, so a room that has a U1 label placed among rooms with the same label allows switching lights and electric power supplies more efficiently.

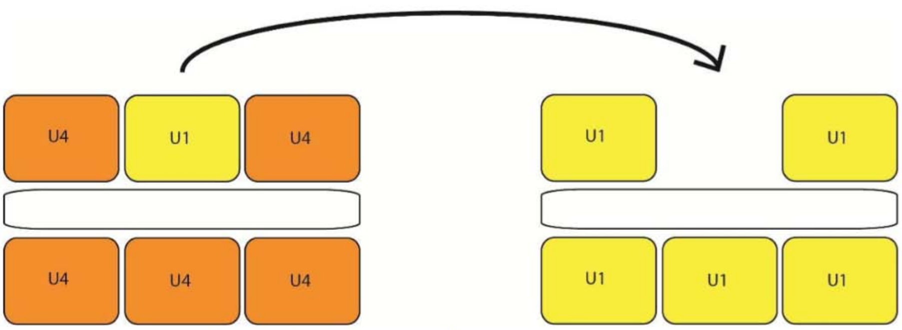

Figure 47: Colour by label helps consider room placement (Streamer D1.6)

4.2.5 Generating design alternatives based on intervention scenarios

The outcome of the fitness analyses of the current layout, the energy performance of the building and the running costs will indicate which intervention type to consider. Furthermore, the changes in organisational needs must be evaluated, for instance a department might be foreseen to grow, might have to shrink or be merged with another department.

Streamer (D1.6) defines as intervention categories:

- interventions on layout and space;
- interventions on building envelope;
- interventions on MEP systems.

A hospital renovation may regard the space layout, the Mechanical, Electrical and Plumbing (MEP) systems and the building envelope, or any combination of the three. The label based evaluation may help to decide whether or not the layout should be changed. If the comparison of function requirements and space properties shows the existing layout to be satisfactory, a choice might be made to only intervene on the MEP systems and/or the building envelope.

The term retrofitting does not limit interventions to the existing building volume. Instead interventions may expand or change the volume. Several intervention concepts are defined in Streamer D1.6 as *Retrofit solutions on a large scale*:

1. Covering an enclosed external space, creating an atrium;
2. Extending a wing;
3. Vertical extension, building on top of an existing building;
4. Adding a temporary building;
5. Adding a second façade on top of the existing façade;
6. Covering current buildings with walls and roof;
7. Placing an extra floor within a high space.

All these interventions take the existing building as a starting point, but modify the volume significantly in ways that change the performance of existing spaces, or that add floor area which creates the possibility for new layouts. Some of the retrofit solutions only improve the building envelope without changing the available floor area (5,6), other solutions increase the available floor area and thus imply rearranging the layout.

Any of these retrofitting solutions will require a human designer to pick a concept for its potential and to modify the constraints in the evolutionary algorithm.

A multitude of retrofitting scenarios may be investigated, intervening on the layout, building envelope and or MEP systems, in combination with any of the large scale retrofit concepts. These may be compared with the non-intervention scenario. The multitude of options will be helped by the Semantic design process and tools.

In comparison to the design of new buildings, design within existing buildings is characterized by additional constraints, listed as:

1. Unmoveable structural elements
2. External walls and roof
3. Fixed vertical elements; shafts, stairs, elevators
4. MEP services, when not subject to replacement
5. Adjacent unmoveable rooms

Adjacency requirements may be defined with design rules for unmoveable rooms that influence the generation of new layouts. Adjacent functions may be inserted as constraints in the EDC by means of design rules.

The possibility to define unmoveable constraints in a plan layout is foreseen in the current development of the experimental Streamer EDC towards a commercial product integrated with DEMO's RE Suite. It introduces a Building Editor environment where the constraints are manually placed, see Figure 48 below.

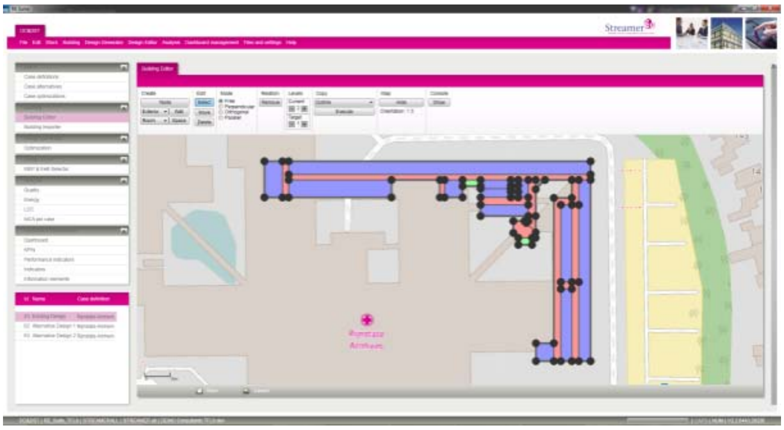

Figure 48: EDC Building Editor for placing constraints (DEMO2017)

Once the plan constraints are defined the application passes on to the automatic generation of space layouts in the Design Generator environment, see Figure 49.

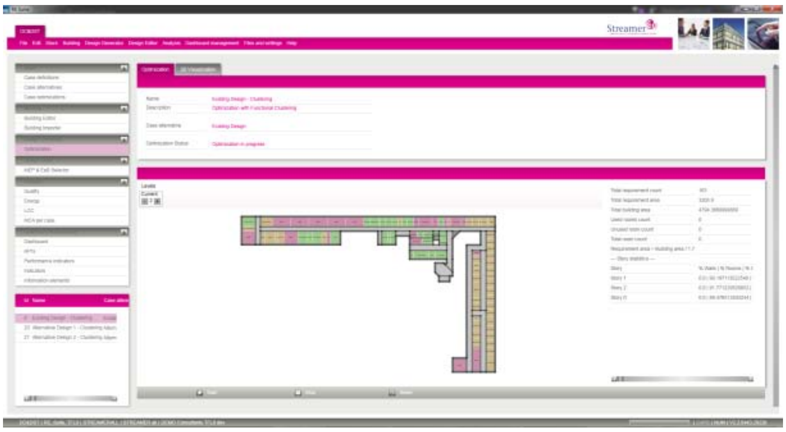

Figure 49: EDC Design Generator automatically placing rooms (DEMO2017)

4.2.6 Evaluating energy performance

The energy consumption profile for each space is strongly determined by its function. A good example for hospitals is the operating theatre with very high ventilation rates, and many high yield appliances.

Deep plan buildings have less surface exposed to heat loss because of a better Surface/Volume ratio. Narrow plan buildings generally have more daylight and natural ventilation, saving on electric lighting and mechanical ventilation.

Ventilation with a constant air flow rate does not adjust flow rate to occupant presence it thus also consumes for heating and cooling. For improved energy performance, this should be replaced by Variable Air Flow ventilation. Local ventilation units to should be centralised ventilation, as larger units are more efficient. Natural ventilation should be used when and where possible. For most functions daylight should be preferred over artificial light.

Consumption for electrical appliances can be reduced by switching them off, the Streamer UserProfile label can help to cluster spaces based on the active hours. The Comfort and Hygiene labels can be used to cluster spaces that may have similar climate control, this can help reducing MEP system length.

For each different retrofitting scenario energy consumption may be calculated. As the alternative retrofitting scenarios have semantic labels, the simulation is no different from the process for new buildings, see paragraph 4.1.9.

4.2.7 Comparing design alternatives with the decision support tool

In the case of retrofitting it is possible to analyse a level of detail which remains unfeasible in Early Stage design for new buildings, e.g. the quality of views from windows or the finishing level.

In all cases the alternative interventions should be compared with the baseline model, which may be the existing building without intervention.

4.3 Semantic BIM Design method extended to selected building types

The Semantic Design Method takes a functional approach to the design of building layouts. It considers first the functional requirements of individual spaces. The exterior, and the relation of interior to exterior are of secondary importance. A pioneer of this approach was Hans Scharoun, that compared the building plan layout to an organism; it is what led him to “organic” shaped plan layouts (Jones1995). The functional approach was systematically elaborated for all typologies by Ernst Neufert in his manual Architects Data (*Bauentwurfslehren*). To discuss the suitability of the Semantic Design Method for typologies other than the hospital, a selection of typologies is derived from the index of Neuferts Manual (see Table 14). Chemical facilities are added because they offer interesting possibilities to discuss the Semantic Design Method despite not being listed by Neufert.

Table 14: Selected building typologies, after Architects Data, Ernst Neufert

CATEGORY	SELECTED BUILDING TYPOLOGY	PARAGRAPH
Residential Buildings	Collective Housing	4.3.1
Accommodation	Prisons**	4.3.2
Education and Research	Schools	4.3.3
Administration and Offices	Offices	4.3.4
Industry and Trade	Chemical Facilities	4.3.5
Health*	Hospital (*see chapters 4.1 & 4.2)	-
**) not present in Neufert		

For each of these selected typologies, the following points will be discussed:

- What are *function specific requirements*?
- Are *custom labels* needed or do existing labels need to be reinterpreted?
- What *design rules* might be formulated for each function?
- Which *performance indicators* are suitable?
- Discussion of a designed example of the typology, as a *limit case*.

For each typology, a contemporary example is treated: the cases are selected for reformulated briefs leading to extreme design solutions and/or integrated volume/floorplan solutions. The discussion regards if, and under which conditions the Semantic BIM design method would be able to find the same solution: With human intervention, re-interpreting the brief or the volume; adding labels or design rules.
The volume is a top-down constraint, the brief, PoR and design rules are bottom up constraints. Does a given solution require simultaneous adjustments of both top-down and bottom up constraints? Simply put, can the computer think out-of-the-box or does it require the human designer?

To understand whether the Semantic BIM design method would allow finding an adequate solution let us look at a case study for each typology, where part of the designed answer lies in the formulation of the question.

Out of scope constraints

Some aspects are out of scope of the evolutionary design algorithm. All the top-down constraints, orientation, floor plan depth, Surface to Volume ratio are out of scope of the algorithm. In the discussed method, these design decisions are up to the human designer. While their impact on energy consumption is significant, they are not subject to optimization.

Density and urban form lie outside of the scope of the described semantic design method. They can be studied separately and in an automated manner generating alternatives within the context of local building regulations as done by Uytenhaak, or based on daylight requirements as done by MVRDV.

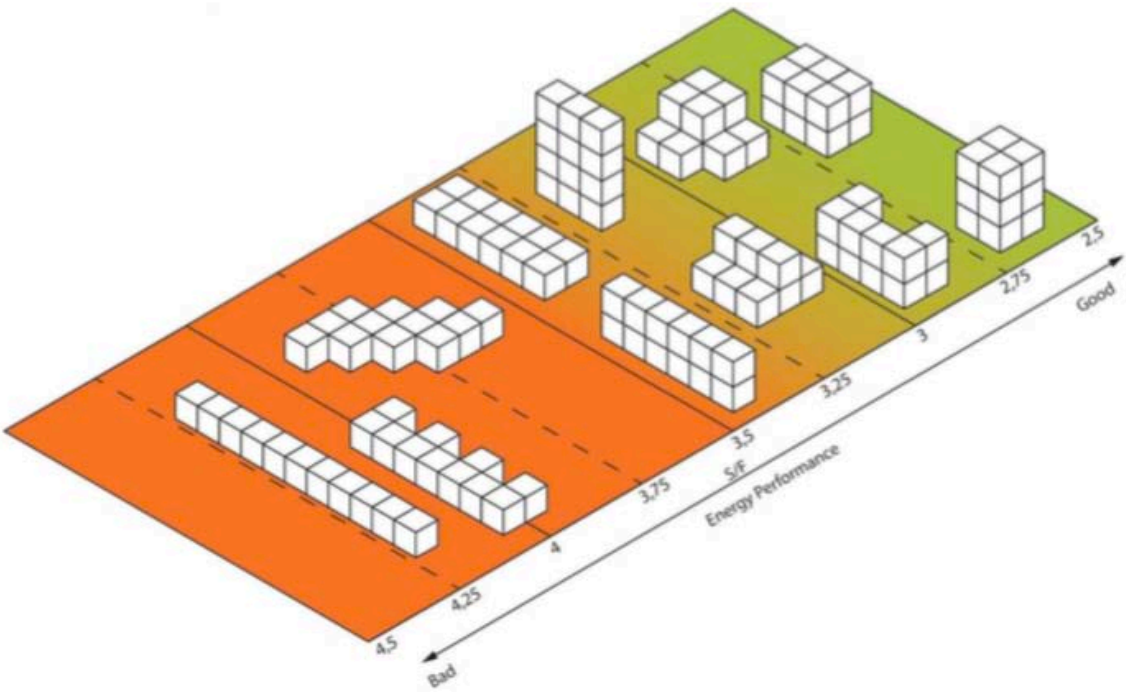

Figure 50: Energy performance in relation to form factor, from Streamer D1.6

4.3.1 Collective Housing

Function specific aspects

The typifying, if not function specific aspects of collective housing can be found as classifications in housing manuals (Moza 2006, Schneider2011). There is an interrelation between density and urban form, access typology and floorplan. Other factors are the relation of public to private, target family size, social housing or market sector.

Of these, density and urban form lie outside of the scope of the EDC. Access typology can be partly determined in the Programme of Requirements, in combination with the design rules; for instance, considering all accessed units together, a corridor or gallery access should consider the entire block, a portico access only those units accessed from the stair.

Space requirements are generally less defined for houses than for hospitals, the possibilities of a chosen bay width and a plan depth might determine the room size to a

large degree. The PoR should therefore contain some range for allowed room dimensions or a minimum width per room type.
Deeply rooted cultural preferences determine how an apartment should be accessed. Within floorplan of individual apartment unit, plan distribution and access are relevant, more so when housing larger families, houses with guest quarters or service staff. Preference for unit access to living room or a through a hall can be indicated in the PoR. The need for a corridor can be expressed in design rules for the room relations.

The preferred mixing or separating of different apartment sizes can be controlled by assigning the apartments to a functional area.

Custom labels

Without defining new labels, The UserAccess label might be used to distinguish public, collective, private zones within the house. The ComfortClass label could be used to assign the privacy requirement to individual rooms.

Design rules

With a requested room area, room width should be maximized because it limits furniture placement. Preferred room relations (cultural aspect) can be expressed as design rules based on the PoR. Preferred room orientation design rules might express that each room type has a preferable orientation. This would require a new design rule type as orientation was not yet expressed. For privacy, an acceptable view distance might be defined to other housing units. Spatial quality and flexibility of use are both partly defined by room relations but also depend on choices that can be made by the designer on a lower level.

Performance indicators

Key performance indicators for collective housing can be maintained as Energy, Cost, Quality. The Quality KPI can be composed from the indicators: distribution of gross to nett area, presence of collective spaces, quality of functional relations, room width as a measure for ability to furnish, quality of views and privacy, spatial quality of the access and the floor plan. As in the case of the hospital some of these aspects are hard to qualify, let alone quantify. So, the selection between design alternatives in this case will have to be done by the architect.

While daylight access is solved by façade adjacency like in Streamer method, privacy and views are not. For layout design these are crucial aspects as the following case study will show. Only some vector based method would solve this, like those that have been developed for the view requirement in LEED (isovist or viewshed, e.g. Heumann, 2011) The need for privacy could be expressed by the ComfortClass label.

Limit case: Cruz & Ortiz on Java Eiland, Amsterdam (NL)

In the case of the urban plan of Sjoerd Soeters for Java Eiland, different architects were given several repeating building blocks of 5 bays of 5,4 meters width to design for a total width of 27 meters, the depth of the entire building was set at 13,5 meters. At some point the density needed to be increased. The office of Cruz & Ortiz designed a solution in which the middle of their block got an appendix projecting 9 meters on the rear (courtyard) side, see Figure 51. The internal balcony while outside of the façade line, sits in the corner, a position reducing indiscrete views of neighbours. Further windows are located on the furthest façade, parallel to the main façade for privacy.

The decision of adding volume and solution for the floorplan are intrinsically bound. The choice to expand the volume is out of scope for the EDC, so either would have to be a human designer that would have to input a volume in the software or the software would have to be able to expand the volume according to the logic the designer uses.

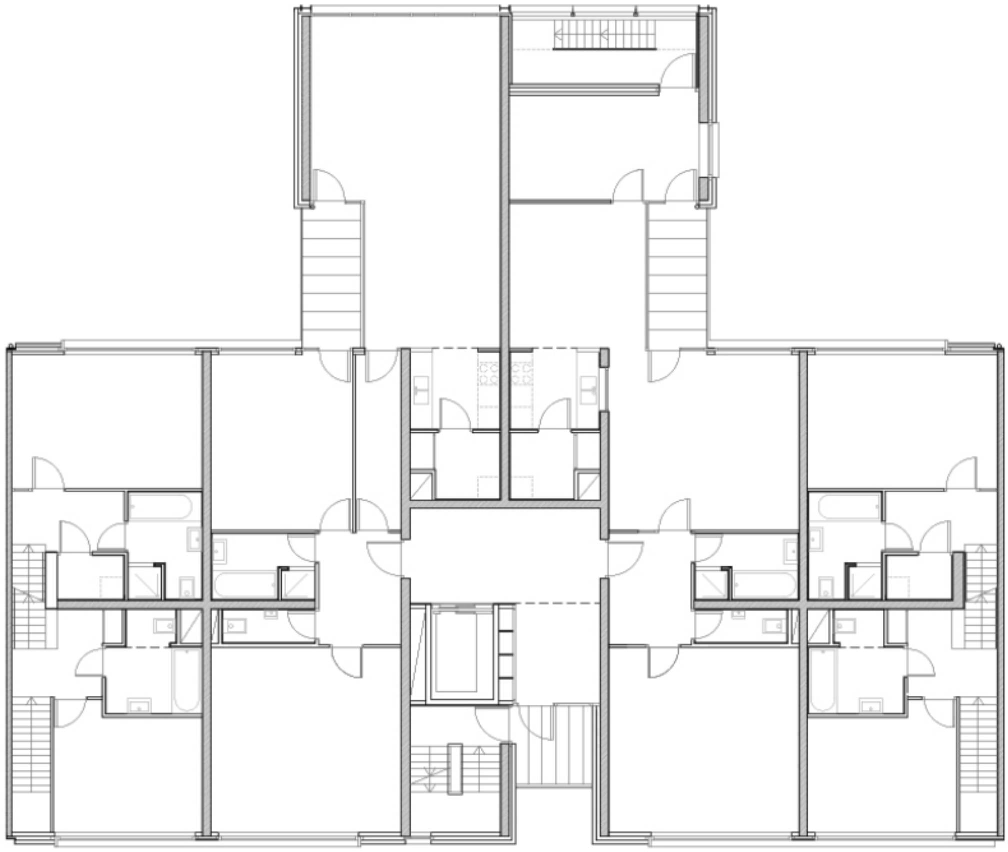

Figure 51: Increased density hammerhead typology, Cruz & Ortiz, Javakade

4.3.2 Prisons

Type specific aspects

The brief for a prison is highly regulated, the space requirements defining the PoR are standardized on a national level. There are strict rules for routing and access that are needed to guarantee the safety of staff and detainees. This building type has zones with different levels of access for staff, visitors and detainees. The means to control these access zones include the topology of spaces, door typologies, surveillance with presence of guards and cameras. Furthermore, camera's, sightlines and human surveillance would also need to be defined.

On a lighter note, a prison bares similarity to Hotel layer in the NHBI approach. The similarity to hotels goes further, as both prisons and hotels can optimize the scale of units to facilities, which has cost aspects such as cleaning, maintenance, being permanently staffed. Both also periodically deal with empty units.

Custom Labels

The UserAccess labels need to be re-interpreted more strictly for access security zones for public, staff and detained.

Design rules

To design with access restricted zones and surveillance, the BIM model would need to contain doors with the possibility of restricting access and visibility aspects either directly through human presence or indirectly by means of cameras. To automate the design doors would have to be distinguished in classes of access restrictions. Design rules would need to be written that place these components, for a functional access restricted design to emerge.

Performance indicators

Functional aspects will be judged by the design validator based on the priorities assigned to design rules. An extra performance indicator might be the energy use also in relation to partial occupation. Quality may be partially measured, and will have to be partially controlled by an expert designer. Surveillance and safety aspects will need to an in-depth analysis which should produce some numeric indicator for comparisons. Prisons should entertain their detainees also during the daytime so the day areas, while restricted, will need to offer some quality.

Limit case: Hootsmans detention centre, Breda (NL)

The example is a youth detention centre by Hootsmans with a gross floor area of 3.870 square meter over three floor levels. It has a cross shaped symmetrical layout, see Figure 52.

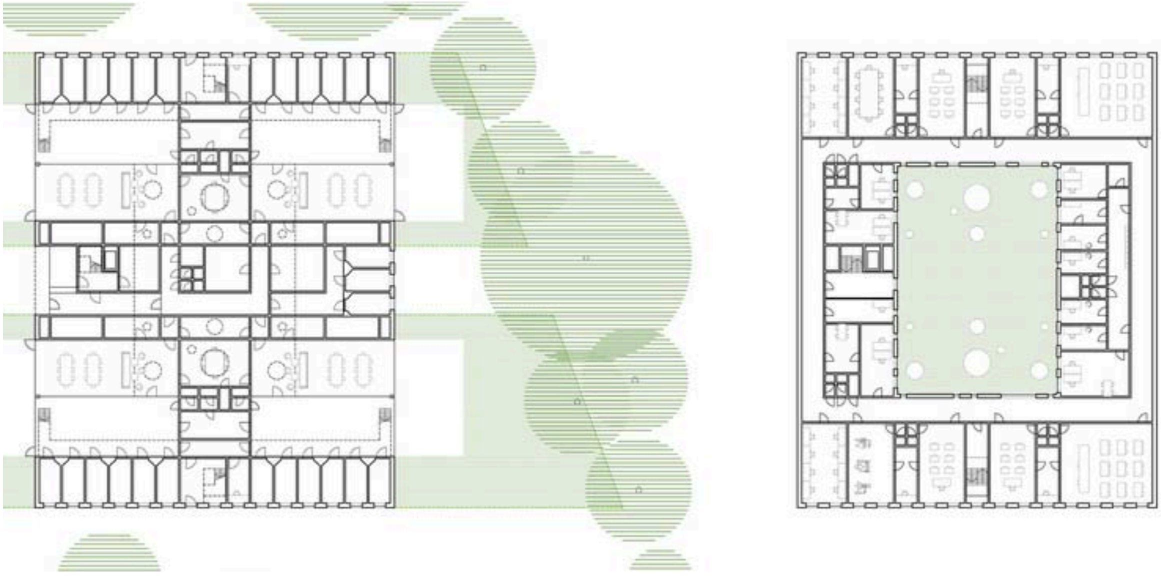

Figure 52: Youth Detention centre Breda by Hootsmans

The ground floor and mezzanine are divided in four equal sectors two for boys and two for girls. Each has a large living room with kitchen and sitting areas, twelve bedrooms over two levels and a bathroom-washing area. These sectors are positioned on the four corners of the building, the living rooms with large windows onto the garden. The central cross contains the rooms for the staff rooms, service and distribution spaces. The mezzanine plan has not been published but can be inferred to consist of staff only spaces, connecting the four sectors and all floors. The top floor has office spaces and meeting rooms on the perimeter, partly inward facing and partly outward facing. The centre is left open to an inaccessible roof garden with round skylights for the spaces below, all windows facing this garden are round too.

This building was selected as a case study for its clear spatial structure, even if it contains the ambiguity that arises from the position and role of the skylights in the roof garden. The option of skylights as a means to supply daylight was not foreseen in the design rules so far. In fact, when a space requires views, skylights are not sufficient, but for certain space types without prolonged presence of people (e.g. meeting rooms) skylights may be a good option.

A lot of the functionality and spatiality of this detention centre derive from the choices at the level of the volume and of the position of the corridor. Both these aspects are out of scope for the evolutionary algorithm (EDC), they need to be initiated by a human designer. Once both the volume and corridor position have been set, the position of rooms can be proposed by the EDC, and if the latter considers the potential of skylights the same layout could be generated.

4.3.3 Schools

Function specific requirements

Collective spaces such as playground, atrium, stairs and corridors fulfil an important social role in everyday life but also may be used by all students and teachers in case of events. The visual and functional relation to outdoor space is important. Corridors vary from being full of students, to being empty with the occasional student. When the school hours are over, peak numbers of students collect at the doors. At times in the corridors, there may be a need for surveillance by teachers as well as a risk of students get distracted by passers-by. Classrooms are often deep spaces, in order to accommodate large groups of students, requiring high ceilings to be daylit. The building needs to meet a high demand of spatial quality. The school takes up an important role in the transfer of culture and its architecture is expected to express this.

Custom labels

While most space requirements for schools seem covered by the Streamer labels, labels to classify outdoor areas might be added, such as a requirement for the time of day they receive direct sunlight.

Design Rules

Once the classroom proportions are determined, the architectural assignment will be a lot about the collective spaces including the corridors and outside spaces. To cut costs, recently some briefs have eliminated the school auditorium. In this case, events will take place in the gym, or in an atrium space, or at the foot of some stairs. How to create collective spaces when not part of the PoR? Design rules might be try to reduce corridor length, while maximizing their width. To obtain an auditorium or atrium like space, a building envelope might be drawn that cannot be filled with classrooms, leaving an undesignated space in the centre.

Performance indicators

Again, the quality performance indicators need to be judged subjectively. How to classify sociocultural component? How to evaluate the social potential or the spatial quality of collective spaces? It is important to have corridor spaces that can have other roles, that may double for events, that have locker or wardrobe spaces, play areas or study corners depending on the age group. The gross/net proportion may disqualify a design alternative which creates a lot of collective space. Some architects created a compact volume, with more gross floor area and have in compensation reduced the external façade area, e.g. Kempe & Thill. A view analysis might be considered to compare collective spaces.

Limit case: XDGA stacked school, Gent (BE)

If we were to take an exceptional volume such as the Oude Dokken school by XDGA, could it conceivably be designed with the semantic BIM method? Which steps might have been made to reformulate the PoR, design rules or the accepted outcomes as measured in performance indicators.

The urban plan prescribed a public passageway through a site for schools with playgrounds the sum of which couldn't conventionally fit on the available plot. XDGA's design solution combined two radical interventions, the first to collect the programme of primary school, after school, nursery and sports facility in one densely packed building, overlapping the shared spaces. The second to stack the playgrounds on several levels allowing the public passageway to pass unhindered below.

The impossibility of locating all play areas on the ground floor is evident from the available plot area. This immediately clarifies the need to stack the programme; rooms, play areas or both. This example would require defining the outdoor play areas a programmatic element that may be stacked. Outdoor areas would have to be added to the PoR for the algorithm to be able to consider them. If this was not done, we could improvise, by treating these areas as a special room type (with walls without performance on their boundary).

Another passage might be in interpreting the PoR to allow some of the collective spaces to be combined in a single space. One way to work towards this result could be by using a design rule that requires them to be adjacent.

So, to conclude, it is possible to use the evolutionary algorithm to design this stacked school. Even if the main design intervention, the mass/void distribution would have to be thought of by the designer once the envelope has been designed manually, even if relations with outdoor spaces would probably have to be obtained by forcing the limits of the labels and evolutionary algorithm.

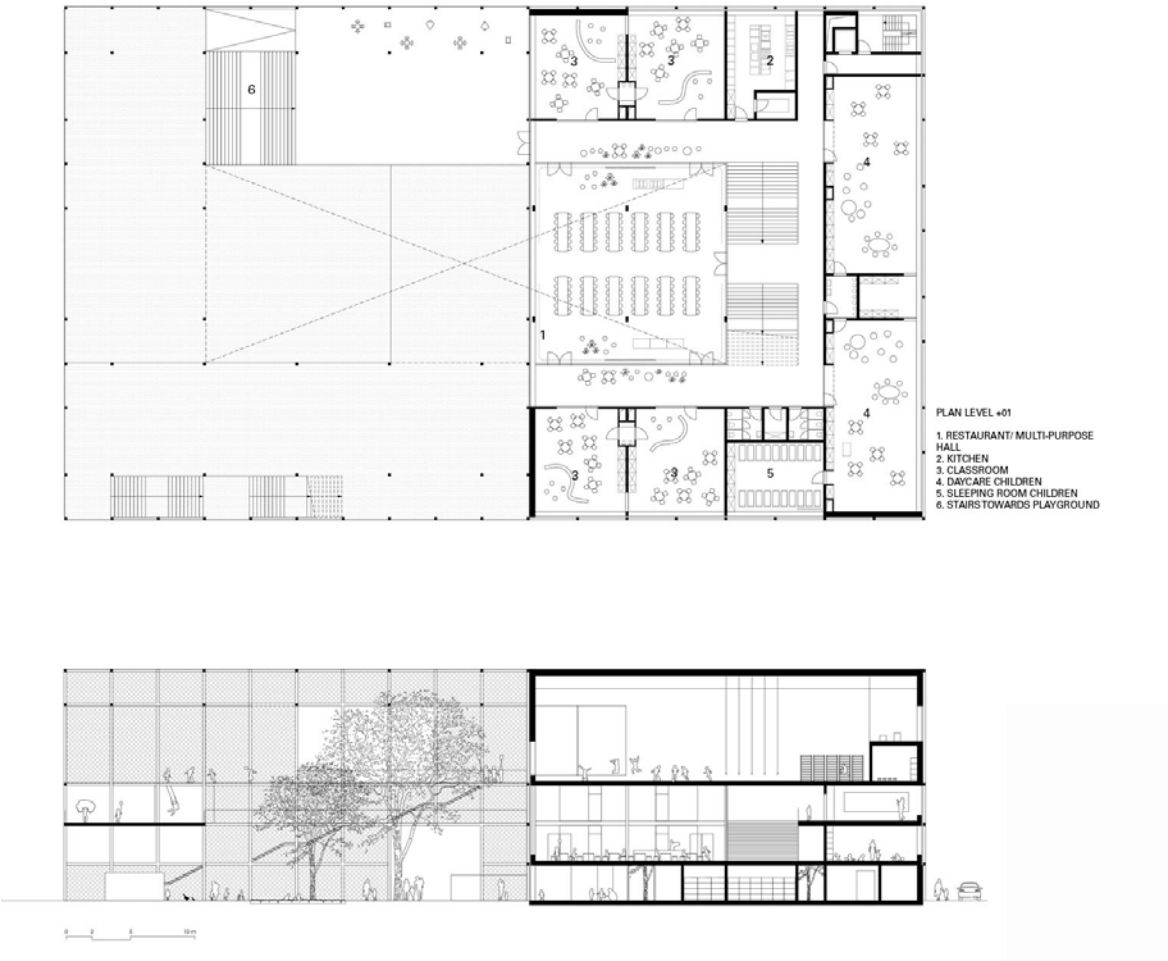

Figure 53: Stacked playground (l), densely mixed school (r), Oude Dokken school, XDGA

4.3.4 Offices

The office typology is already classified and described in the Streamer methodology as one of the NBHI layers. The main space type will not differ substantially in its requirements from the office spaces in a hospital. Generally, office spaces and office PoR are quite homogeneous, the main choice is between an open plan office or a cell office. In open spaces, individual needs such as privacy or comfort are more difficult to obtain. In addition to main office spaces (whether rooms or open space), there will be several meeting rooms, services and some circulation area which may be combined with collective space.

Custom labels

When a client is known, a PoR will mention departments that need to be related. In many cases however, offices are built without knowing the client. If this is the case, the rentable unit size is an important measure. In either cases, departments or rentable units can be grouped with the functional area label. There does not seem to be need for custom labels.

Design rules

Let us consider design rules in relation to the minimum space unit. The minimum unit in an open plan situation is a desk or group of desks in a larger space. In the context of the EDC, the question could be how to deal with this unit as an input. We could think of it as a zone, or as a room without walls. Neither of the two options have been considered in the EDC as it works at a slightly higher scale level, as the minimum spatial unit is the room. We could set separation walls to be defined without any performance. The following case study will show how defining the modular aspect can be.

Performance indicators

Presuming that the KPI's for energy and cost apply in the same way as in the case of the hospital, the question is how to define the quality KPI for an office building. Quality performance indicators might include flexibility resulting from modularity and divisibility, the balance of control of individual comfort in a collective environment, comprising perceived physical comfort through climate control and visual comfort i.e. uniformity of daylight and absence of glare. These aspects are all quite technical and can be predicted with manual architectural modularity studies, dynamic thermal simulations, climate based daylight studies. So, while not yet foreseen in the Streamer case for Early Design, the predictability is quite high. A sequence of automated simulations could allow comparing early design alternatives as described above.

Limit case: Campo Baeza, Bank, Granada (ES)

As case study office have a look at the typical floorplan of the Caja Granada bank office, by Campo Baeza, shown in Figure 54. The given example is chosen for its architectural strength and apparently rational layout, which would seem to lend itself to computer aided design. A close look at the plan reveals a strong modularity. The grid size is three by three meters, each desk is inside of one module, two elevators also share one module. The stairs and toilets share two modules, the corridors are one module wide minus one or two storage cabinets. The wings are either double loaded with an external sun screen for a total depth of six modules, or single loaded for a depth of three modules. The centre leaves an atrium void of ten by ten modules within which four hollow pillars are located.

While the publicly available information does not mention it, most likely an articulation in departments was requested by the client. The plan does not seem to be overly influenced by articulations in the programme, it maintains a generic appearance; each one desk module equals any other desks module. Some meeting rooms and tables are inserted without upsetting the scheme.
Looking at the proportions in plan, about forty percent is composed of actual office spaces, nearly a quarter are modules dedicated to corridors, while the central atrium takes up over a quarter of the plan area. Significantly, two space types not typically requested in a PoR, corridors and atrium void space together account for over half of the plan area. Here lies the first human design choice. To obtain a design like the example, either the atrium void would need to be translated into a PoR input for automated design. Or the building shell would have to be drawn as an exterior courtyard.

The clarity and rationality expressed in the layout seems to be the result of the rigid modularity. This is a design choice which lies outside of the scope of the EDC and its design rules. An analysis could break the PoR into component spaces (desk, meeting space, corridor, stair), then a manual design of these component spaces might suggest which module size to adapt, however this is outside of the scope of discussed semantic BIM method.
Design rules could be developed to weigh modularity, for an open plan it might be helpful to treat the single desk as a small space type in the PoR, otherwise it will remain out of scope for the design automation.
The different building depth in the two wings in the EDC is a manually applied top down constraint, so the choice in the case of the EDC is human. Generating different design alternatives should be quicker than in a manual process, so comparing different volumes should be easier than in a traditional process.

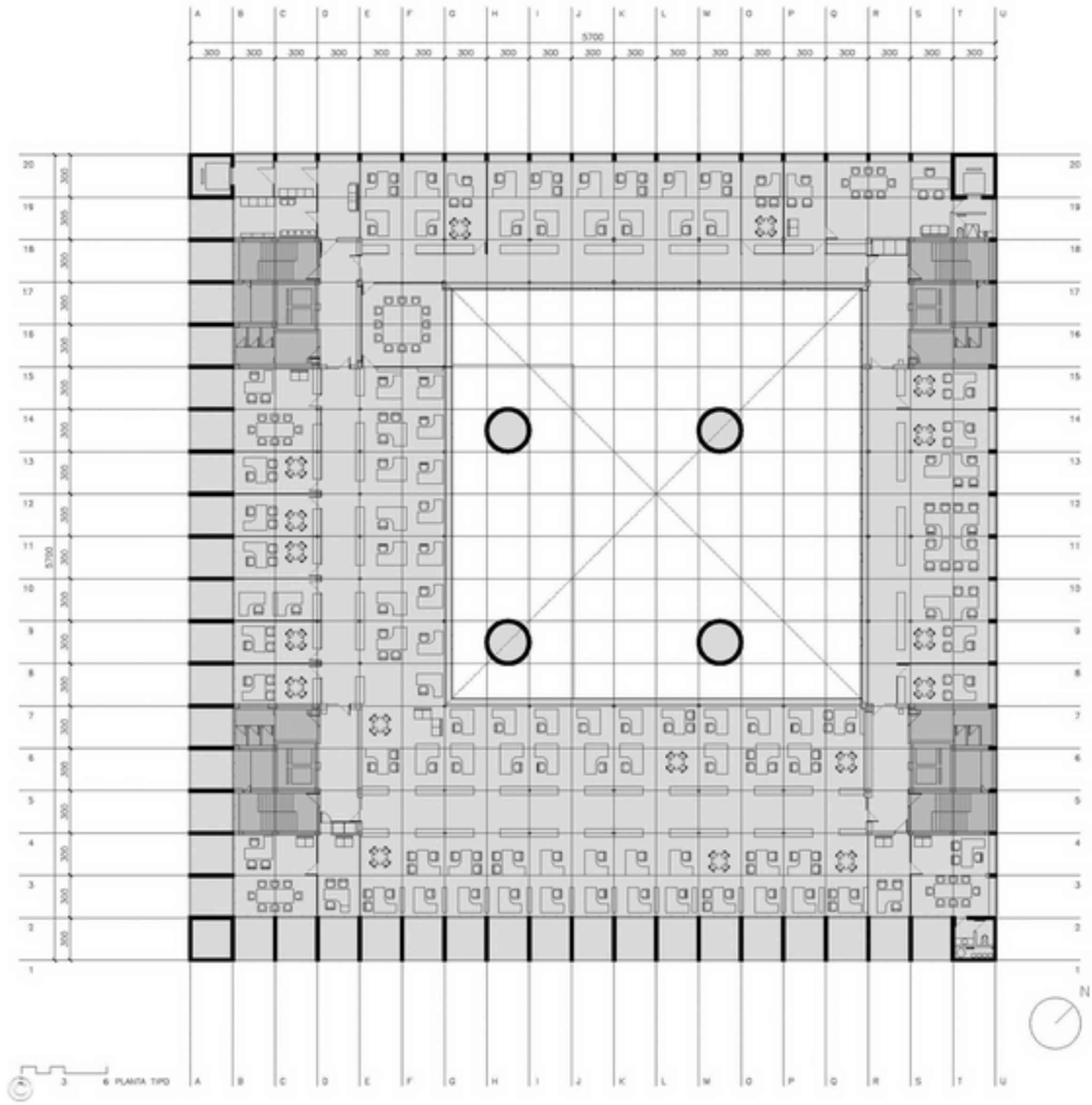

Figure 54: Typical floorplan of the Caja Granada Bank, Alberto Campo Baeza 2001

4.3.5 Chemical Facilities

Function specific requirements

Computing with the evolutionary algorithm is facilitated by clear rules and procedures. Chemical processes are influenced by volume, flow direction, gravity or heat/cold induction. For an industrial chemical plant process information should be well known and readily available. With current techniques, chemical processes can now be extensively simulated. Outside of the architecture domain, electronic circuit design and piping design are already subject to design automation, see chapter 3.3. A chemical plant can be thought of as a complex special version of a circuit.

The layout must be purely functional determined by production requirements. A building envelope might either be built around the machinery: first the chemical plant, then building envelope. Or the building envelope is a constraint on the extension of the plant. Either way a layout will have formal characteristics influencing plant performance. Pressure loss occurs due to pipe length, diameters and curves, calculations will be necessary to verify each generated design. Another aspect is safety, there will be minimum distances between substances and between chemical processes, hazards may include explosion, spills, minimum distances for human operators and safety distances that need be respected on an urban level.

Custom labels

Labels could be developed based on substance, process, temperature. The chemical substances contained in vessels and piping could be classified as were the RoomTypes in the Streamer process.

Design rules

Design rules could be described which optimize all the chemical processes for each reaction in the production chain. The building envelope may or may not be a constraint. Risks could be minimized by guaranteeing distances. The chemical substances contained in vessels and piping could be used to control adjacency, reducing risks of breakage.

Performance indicators

Investment and running cost are performance indicators for the lifecycle cost KPI. Plant efficiency might have as indicators energy consumption and production rate. A new KPI might be safety; composed of risk, and failure rate.

Since subjective indicators such as architectural quality are not requested, the chemical plant would appear an ideal subject for design automation. Granted that it lies outside of the architecture domain, it also lies outside of the authors field of competence. No example shall be discussed.

5 Discussion and conclusion

5.1 Summary of main findings

5.1.1 Philosophy of the method

The author has described a process in which an architectural brief is clearly defined in requested spaces with classified labelled requirements and design rules describing room relations and normative constraints. This brief, with manually applied urban and architectural level constraints, is used as input for an evolutionary design algorithm that generates alternative design options. The designs are then validated before undergoing energy and cost analysis. Finally, an analysis visualisation tool helps choosing which design shall be selected for the next phase.

The extension of the method from new hospital buildings to renovation of existing buildings and other typologies further investigate the scope and limitations of the method.

The semantic BIM design method imposes a clear definition of constraints: space requirements, design rules and external constraints. The method requires explicit conditions to be defined. This in itself may be seen as an advantage because the process becomes more open and transparent. The human designer interprets the brief, a certain freedom in applying the available labels is necessary to deal with functions other than hospitals. The same can be said for the design rules, the priority of which is arbitrary. The design rules need to comprise all the fundamental constraints. Since the algorithm has no experience it will not automatically consider good practice as a designer would.
There emerges a distribution of roles between human designer and evolutionary algorithm.

The semantic BIM design method appears to be more suitable for well-defined programs with many different space typologies, each with specific requirements. And possibly topological requirements; such as corridors with specific access requirements. Examples are the hospital, the detention centre, even the chemical plant.

The method seems less comprehensive for buildings with less stringent spatial requirements, poorly defined assignments with a large cultural component; theatres, schools. Solutions will be proposed in any case, but a human designer can improve the solutions by intervening in the interpretation of the brief and the definition of the constraints. An example of creative interpretation of the brief is the atrium as a space requirement, both in the Campo Baeza bank and XDGA's School building, see chapter 4.3.

The examples for collective housing and school typologies have shown that in some cases outdoor space need to be designed together with indoor spaces.

The process keeps on adding data, avoiding data loss by writing it to the same open standard IFC file. This data is then used in every next step. As the design option is evaluated, information from earlier steps is used to inform successive steps. The enrichment of a single open standard IFC file allows this. Applications are required that can import this information from the IFC file.[5]

[5] Alternatively, domain specialised applications can share information in an online database e.g. zapier.com, flux.io (Van Berlo, Streamer D6.6)

The evolutionary algorithm will consider both top-down and bottom up constraints but in a random, heuristic manner. This means it will visualise solutions that at an experienced designer would discard at a first glance. Only over time will it discard these solutions if a correctly prioritized set of design rules indicates so. This may be slower than an experienced designer working in a structured manner. Examples of structured design steps a human designer might make are the use of functional areas to analyse grossly proportions to find possible room clusters to design with, or the detailed analysis of parts of the PoR to decide on a suitable modularity. On the other hand, the evaluation of many generated design options limited only by calculation time could allow optimizing complexity of requirements.

The examples for collective housing and school typologies have shown that in some cases outdoor space need to be designed together with indoor spaces.

5.1.2 Constraints are not subject to optimization

Design is facilitated by constraints. In architecture (education), the hardest assignments are those without constraints. In that case, there are too many options to consider.

The manually determined building depth is a constraint that will largely determine possibilities for room placement within the floor plan. This suggests, as does the example of the modular office in paragraph 4.3.4, that it is still useful to make floor plan studies to determine desired modularity before running the EDC.

Aspects that are not subject to optimization miss out on a part of the potential of the reasoning. Example of the corridor position in an earlier version of the EDC and the final released version… In the earlier version, the access typology is chosen but the algorithm stretches it to fit: “For each part of an empty building shell, a first layout is randomly chosen from a database of layout templates. These layout templates determine how corridors are to be placed, and how the rooms are aligned along these corridors in strings of rooms.”

In the final released version the building editor gives human designer control over corridor placement in building depth. The increased control of the designer reduces the possibility of the algorithm to suggest optimum corridor placement in relation to the room proportion (width-length).

5.2 Strength and limitations

5.2.1 Limitations of the human designer

On complex building assignments, human designers work in a design team, each individual makes design choices based on personal reasoning or preferences. Design teams will change over the course of the project so the basis of certain design choices may not be clear to all involved, in particular the client. For building projects that consume huge financial resources it may be desirable to be able to follow the process of the design lifecycle. The described methodology allows this because space requirements and design rules will remain explicitly available throughout and beyond the design cycle.

The human designer cannot consider too many constraints simultaneously. The evolutionary algorithm generates infinite series of designs and rates them, allowing all constraints to be rated in the order of their priority.

5.2.2 Retaining explicit knowledge

Strengths of the described method include that it requires the client to explicitly define programmatic requirements. The consultants are forced to lay down explicit design rules. Both are retained throughout and beyond the design and building process allowing for transparent review of the process and its input. Different architects make different choices when confronted with the same design problems, as based on personal or cultural preferences. The choice of design rules and their priority will allow an observer to recognize these preferences. Within the semantic design method, establishing the rules and their priority becomes a creative choice.

5.2.3 Automate repetitive work

Dealing with many conflicting constraints and possibly an over constrained assignment lies perfectly within the application scope of an evolutionary algorithm. As the description of the EDC shows it is a packing algorithm which leaves a lot of control with the designer to define constraints. Packing or distributing the single spaces is a quite a complex tedious task with many hard and soft constraints. It can be argued that the distribution of most (supporting) spaces does not significantly alter the architectural experience of a building. Those important spaces the architect does want to place, in the EDC can be locked in a chosen position.

5.2.4 Illusion of the optimum solution

Limitations to the proposed method include the difficulty of finding the optimum solution. The use of an evolutionary design algorithm finds a solution over time based on the defined priorities. The definition of the optimum (each design rule is weighed by its priority), the large amount of spaces, with an exponential number of possible layouts mean that the best solution may not be found, or may have not have been recognized because of the arbitrarily assigned priority of the design rules.

In fact, many arbitrary decisions will still influence the result. Imagine the decision maker simply changing the weight of performance indicators. Even the seemingly more objective information such as the results from energy simulation is unsure, for many reasons:

- because the simulation models give very different outcomes based on the same input
- because they are simplified models
- because users behave in unexpected ways
- because unlike the weather they are based on normalized years

- because of climate change which means that historic weather files are losing their predictive value[6].

Performance indicators regarding quality such as views, or perceived quality of the environment, while having a large impact on the user experience, can be impossible to simulate in the preliminary design phase. The quality aspect risks to be neglected as it has no numeric indicators, it needs to be arbitrarily evaluated by the involved designer. In confrontation with energy and cost indicators, supported by calculated values, this is a weak position.

5.2.5 Design responds to a poorly defined problem

Most architectural assignments and their briefs are not as well defined as hospitals are. Where the brief is less clear and only a limited number of design rules can be implemented this will lead to poor design results. According to Schneider et Al. simply increasing the rules won't solve this:

"A problem is operational if it can be described so accurately that one can specify the steps necessary to solve it. ... The goal of the analysis of a design is a description that is so accurate that it contains the solution, definable and tangible criteria for describing problems are referred to as operational criteria.

…

Architectural design problems are usually non-operational problems and accordingly differ from most problems in other engineering disciplines. The answer to such questions invariably depends on intuitive, subjective and contextual aspects. Creative decisions are always a response to poorly defined situations and solutions are always a product of both operational and non- operational issues."

…

"The goal of the analysis of a design is a description that is so accurate that it contains the solution, definable and tangible criteria for describing problems are referred to as operational criteria"

(Schneider & Fischer 2010, p369-370)

In response to this question, the described method and the limited scope of the EDC as a packing algorithm are advantages, as they allow the human designer to interact in several ways. Redesigning the building volume, changing the design rules (or their priority), blocking spaces in a certain position.

5.2.6 Design uses intuitive, non-rational methods

"… we need to be aware that designing is a process that occurs at different levels and degrees of abstraction. The solution space is explored in the realm between intuition and

[6] Recently weather patterns have begun to shift from climate as registered in weather files. The Dutch meteorological institute KNMI has released an adjusted weatherfile which has more likeness to current weather than the historical file. The weather file takes climate change into account by raising all in which all temperatures by 2°, The use of this file gives better predictions of cooling and heating demand than the standard one based wholly on the long-term average.

rationality in a variety of ways. Good solutions can only arise through an intensive and fluid dialogue between the designer and the generating system.”
(schneider2011rethinking)

The hands-on assisted design process as described does not exclude any of creative methods such as association, intuition, scaling and inversion, unlike some of the automated layout generation processes described in chapter 4.2.5.

5.3 Conclusions & suggestions for further research

In the described method, urbanistic constraints such as the building volume are applied top down. All room properties are applied bottom up. Another aspect of design is the interaction between scale levels. How can these interact?

The described methodology for new buildings fits a layout within a predefined building volume. The manual functionalist approach (e.g. Scharoun) would start out articulating the functional relations between spaces and determine the volume from the inside out. While this might not be ideal for many reasons predefining the volume is a very strong constraint especially on a building that must follow the functional aspect such as a hospital.

Paragraphs 5.2.5-5.2.6 Are basically about the strengths of a traditional architecture approach. Examples of methods a human designer uses that are not implemented in the Early Design Configurator (EDC, paragraph 4.1.5) are:

- analysing and clustering the PoR in functional areas
- working top-down and bottom up recursively
- using rules of thumb and guidelines

The dumb calculations of the prototype evolutionary EDC, trying out a near infinite number of alternatives, do not seem a match.

While the EDC application is a working software prototype, within the scope of the Streamer project only few design rules have been implemented. The generated designs therefore contain evident shortcomings. Consequently, a degree of imagination is required to understand the potential of the instrument. Beyond the Streamer project, the EDC software development is being continued by a commercial partner, so hopefully in the near future more comprehensive tests may be undertaken.

5.3.1 Iterated interaction between designer and brief

A multi-parameter analysis with an increasing number of variables will create a near infinite range of possible solutions, all of which are impossible to simulate.
The fitness landscape of possible solutions is multidimensional and may be non-linear, discontinuous and thus unpredictable.

The presented Semantic design method will only be successful in a context that is sufficiently constrained: a step by step method, optimizing first the volume, then creatively analysing the PoR in a way that inspires solutions. A circular reiterated approach is not new but deeply grounded in design theory and practice.

Steps in a shared human-algorithm design task might include:

1. Follow a two-step design process with an initial optimization of the volumetric envelope on urban level, followed by a room packing with semantic rule based packing algorithm.
2. Allow the human designer to input pre-structured designs to the evolutionary algorithm. This might use design rules-of-thumb, modularity studies or other domain knowledge to increase the usefulness of the outcome. A combination of design guidelines will reduce the amount of low viability designs e.g. Wall-to Window ratio, Building height to depth.

5.3.2 Reasoning with ifcOWL ontology

The chain from semantic ontologies, creating knowledge bases prepares for artificial intelligence. Linked data outside the IFC domain may be accessed with the ifcOWL standard. This will permit logic reasoning (first and second order inference) and in the long term the application of artificial intelligence.

OWL is particularly suited for reasoning as it defines relationships between concepts, for example:

`owl:SymmetricProperty`

If a wall is an external wall, the window in that wall is an external window, so it follows that if a window is an external window, the wall it is inserted in is an external wall. Once this information has been made explicit, we can evaluate if the wall and the window meet all the technical specifications that exist for an external component, e.g. U-value based on nation and climate zone.

Tegmark describes intelligence as *"the ability to accomplish complex goals".* Finding the optimum layout for a complex building brief might be such a goal.

Extracts from sample file downloaded from Streamer Reqcap portal at (requires user authorization): http://85.10.201.48:4571/en/contexts

The file is exported in Turtle, it starts with a statement of the used namespaces, each identified by a prefix. The ifc prefix refers to the recently adopted ifcOWL standard.

```
### OWL output generated by ReqCap (by AEC3)
@prefix reqcap: <http://www.aec3.de/reqcap/20#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC4_ADD1#> .
@base <http://www.aec3.de/reqcap/20#> .

reqcap:Building rdf:type owl:Class .
   reqcap:Building owl:equivalentClass ifc:IfcBuilding .

reqcap:Door rdf:type owl:Class .
   reqcap:Door owl:equivalentClass ifc:IfcDoorStandardCase .


…

# Exchange: ER2-EDC 02 - Output by Early Design Configurator : S01 Early
Design

reqcap:Accessibility_Labels      a            owl:DatatypeProperty      ,
owl:FunctionalProperty ;
   rdfs:range   xsd:string .

reqcap:Room_and_Room_type rdfs:subClassOf [ a owl:Restriction ;
          owl:onClass xsd:string ;
          owl:onProperty reqcap:Accessibility_Labels ;
          owl:hasValue "54"  ] .

reqcap:Assignment_to_zone a  owl:DatatypeProperty , owl:FunctionalProperty
;
   rdfs:range   xsd:string .

(end of sample)
```

6 Thematic bibliography

6.1 Building Information Modelling

BuildingSmart International. 2007. "ISO 12006-3: 2007 (Building construction: Organization of information about construction works, Part 3: Framework for object-oriented information)."

BuildingSmart International. 2010. "ISO 29481-1:2010 Building information modelling -- Information delivery manual -- Part 1: Methodology and format."

BuildingSmart International. 2013. "ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries."

BuildingSmart Model Support Group 1996-2010. "Industry Foundation Classes Release 2x4 (IFC2x4) Release Candidate 2." http://www.buildingsmart-tech.org/ifc/IFC2x4/rc2/html/index.htm.

Eastman, C. M. and D. Fisher, et al. 1974. "An Outline of the Building Description System. Research Report No. 50." Tech. rep., Carnegie–Mellon University.

Acknowledged as first description of BIM

Hietanen, Jiri, and Sakari Lehtinen. 2014. "MVD and Simplebim®." techreport, Datacubist Oy.

Description of IFC model manipulations and stripping as copy-editing as an adaptable, pragmatic alternative to standardised Model View Definitions

Kensek, Karen, and Douglas Noble. 2014. "Building Information Modeling: BIM in Current and Future Practice." Edited by Karen M. Kensek and Douglas Noble. (John Wiley & Sons) 432. doi:10.1002/9781119174752.

Overview of BIM history, current practice and projected future development

See, Richard and Welle, Benjamin. 2009. "AECOO-1 Testbed Information Delivery Manual (IDM) for Building Performance and Energy Analysis (BPEA) Thread." http://portal.opengeospatial.org/files/?artifact_id=29385.

See, Richard. 2011. "Architectural Design to Building Energy Analysis." May. http://www.blis-project.org/IAI-VD/reporting/listMVDs_4.php?SRT=&MVD=GSA-003&DV=2.

See, Richard. May 2011. "Concept Design to Building Energy Analysis - Exchange Requirements Model

See, Richard. May 2011. "Concept Design to Building Energy Analysis – IFC 2x3 Binding Concepts Model."

Series of technical specifications for the transfer of early design IFC models for energy simulation by the author who went on to develop Simergy Pro.

Sunesen, Steen. 2015. "buildingSMART Norway Guide." http://bips.dk/files/article_files/bsnorge-guide-og-rapporter_steen-sunesen.pdf

Introduction to advanced nationwide BIM standard using predefined Bim-Q process templates and export of mvdXML file verification filters.

Torma, Seppo. 2013. "Semantic Linking of Building Information Models." *2013 {IEEE} Seventh International Conference on Semantic Computing.* Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/icsc.2013.80.

Weisse M., Liebich T. and Nisbett N. 2015. " State-of-the-art review of advancements and challenges in ontology research." Research report D5.1, Streamer consortium.

6.1.1 Tools:

Briefbuilder: http://www.briefbuilder.com/
Bimserver: http://bimserver.org/
BIM-Q: Database accessible at (with user login): http://85.10.201.48:4571/
Bizagi: http://www.bizagi.com/
BLIS project MVD's: http://www.blis-project.org/IAI-MVD/
dRofus: http://www.drofus.no/en/
IfcCheckingTool: https://www.iai.kit.edu/1649.php
IfcDoc: http://www.buildingsmart-tech.org/.../ifcdoc-tool/ifcdoc-beta-summary/
mvdXML Checker: https://github.com/opensourceBIM/mvdXMLChecker
SimpleBIM: http://www.datacubist.com/
Solibri Model Checker: https://www.solibri.com/products/solibri-model-checker/
xBIM: https://github.com/xBimTeam

6.2 Semantics, Ontologies

Berners-Lee, Tim. 2006-2009. https://www.w3.org/DesignIssues/LinkedData.html

Gruber, Tom. 2009. Definition of Ontology in the Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag. http://tomgruber.org/writing/ontology-definition-2007.htm

Häfele, Karl-Heinz and Benner Joachim. 2015. "Advance Mapping Structures and Standards." Research report D6.5, Streamer consortium.

Helm, Pim van der and Léon Berlo 2015. "IFC model checking with N3: a proposal for a semantic rule checking language for BIM" *3rd Workshop on international Linked Data in Architecture and Construction*

Hjelseth, Eilif, and Nick Nisbet. 2010. "Exploring semantic based model checking." *Proceedings of the 2010 27th CIB W78 international conference.*

Maedche, Alexander. 2012. *Ontology learning for the semantic web.* Vol. 665. Springer Science & Business Media. doi:10.1007/978-1-4615-0925-7.

Pauwels, P., E. Corry, and J. O. Donnell. 2014. "Representing SimModel in the Web Ontology Language." *Computing in Civil and Building Engineering (2014).* American Society of Civil Engineers (ASCE). doi:10.1061/9780784413616.282.

Pauwels

Terkaj

Weisse M., Liebich T. and Nisbett N. 2016. Semantic Web based Product Modelling Ontology (PMO) - BIM Information Management and Quality Control." Research report D5.2, Streamer consortium.

Zhang, C.; Beetz, J. & de Vries, B. 2013. "Towards Model View Definition on Semantic Level: A State of the Art Review."

Zhang, Chi and Beetz, Jakob,. 2014. "Model view checking: automated validation for IFC building models." *ECPPM 2014 - 10th European Conference on Product and Process Modelling.* Informa UK Limited. 24-39. doi:10.1201/b17396-24.

Zhang, Chi and Beetz, Jakob, and Bauke de Vries. 2014. "An ontological approach for semantic validation of IFC models." *Proceedings of the 21st International Workshop on Intelligent Computing in Engineering (EG-ICE).*

Zhang, Beetz, Weisse. 2015. "Interoperable Validation for IFC Building Models using Open Standards." *Journal of Information Technology in Construction (ITcon)* 20: 24-39.

6.2.1 Tools

conversion from Ifc EXPRESS to IfcOWL with the following tools:
http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/
https://github.com/pipauwel/EXPRESStoOWL
http://www.terkaj.com/tools.html
Open source rule checker for BIM:
https://github.com/openBIMstandards/modelcheckN3

http://openbimstandards.org/standards/
http://www.essepuntato.it/lode

ifcOWL ontologies:
http://linkedbuildingdata.net/resources/IFC4_ADD1.owl
http://linkedbuildingdata.net/resources/IFC2X3_TC1.ttl

ontology-oriented programming in Python, reasoning via HermiT:
https://pypi.python.org/pypi/Owlready

OWL 2 for reasoning; geospatial location aware graphs:
http://www.stardog.com/

6.3 Evidence based design

Berlo, L. A. H. M. van and Bektas, E., and H. J. L. M. Vullings. 2016. "Using Multi Criteria Analyses and BIM to evaluate design solutions of complex buildings; A case study in automated KPI checking in the early design phase." Edited by Harry Timmermans. *Design & Decision Support Systems in Architecture and Urban Planning.* 2-14.

Cama, Rosalyn. 2009. *Evidence-based healthcare design.* John Wiley & Sons.

Schreuder, E and Schippers-Trifan, O., and Traversari, Roberto. 2015. "Process-oriented EeB KPIs in the operation, maintenance, (re)construction phases." Research report D3.2, Streamer consortium.

Nedin, P. 2014. "Building-orientated EeB KPIs of newly designed and retrofitted buildings." Research report D3.1, Streamer consortium.

6.4 Knowledge based engineering, Design automation, Evolutionary design

Abrishami, S. and Goulding, J. S. and Rahimian, F. P. and Ganah, A. 2014. "Integration of BIM and generative design to exploit AEC conceptual design innovation." *ITcon* 19: 350-359.

Duarte, José P. 2005. "A discursive grammar for customizing mass housing: the case of Siza's houses at Malagueira." *Automation in Construction* (Elsevier BV) 14: 265-275. doi:10.1016/j.autcon.2004.07.013.
Design automation based on shape grammar

Elezkurtaj, Tomor, and Georg Franck. 2002. "Algorithmic support of creative architectural design." *Umbau* 2: 16.
Multilevel design automation with topological and evolutionary mechanisms

Jo, Jun H., and John S. Gero. 1998. "Space layout planning using an evolutionary approach." *Artificial Intelligence in Engineering* (Elsevier BV) 12: 149-162. doi:10.1016/s0954-1810(97)00037-x.
Compares two different methods of evolutionary solvers in layout generation.

Lawson, Bryan R. 2006. "How Designers Think – The Design Process Demystified."

Lorenz, Wolfgang E., Martin Bicher, and Gabriel X. Wurzer. 2015. "Adjacency in hospital planning." *IFAC-PapersOnLine* (Elsevier) 48: 862-867. doi:10.1016/j.ifacol.2015.05.118.
Generation of hospital spatial schemes based on simulation of physical attraction

van Nederpelt, Stefan. 2015. " Parametric modelling techniques for EeB." Research report D5.5, Streamer consortium.

Rutten, David. 2010. "Evolutionary Principles applied to Problem solving using Galapagos." Edited by Cristiano Ceccato, Lars Hesselgren, Mark Pauly, Helmut Pottmann and Johannes Wallner. *Advances in Architectural Geometry 2010.* Springer Vienna.
Clear explanation of possibilities and limitations of evolutionary design by the author of Grasshopper3d and Galapagos

Sleiman, Hassan and Hempel, Steffen. 2015. "Configurator of workflow and building process requirements." Research report 6.1, Streamer consortium. (restricted)
Selection and configuration of design rules as input for automated generation of layouts.

Schneider, Sven, Fischer, Jan-Ruben and König, Reinhard. 2011. "Rethinking automated layout design: developing a creative evolutionary design method for the layout problems in architecture and urban design." In *Design Computing and Cognition'10*, 367-386. Springer. doi:10.1007/978-94-007-0510-4_20.
Describes the design process as poorly defined problem thus not suitable for automated solving without real-time interventions of a designer.

6.4.1 Tools

Autodesk Akaba: https://redshift.autodesk.com/make-your-own-rules-with-generative-architecture/ available on request from: lineshapespace@gmail.com
Eclipse: https://eclipse.org/
Galapagos: http://www.grasshopper3d.com/group/galapagos?
KIT Early Design Configurator:

6.5 Early stage energy simulation

Cemesova. 2013. "Enhancing BIM-based data transfer to support the design of low energy buildings." Ph.D. dissertation, Cardiff University. http://orca.cf.ac.uk/id/eprint/57565

Jalaei, F. and Jrade, A. 2014. "Integrating Building Information Modeling (BIM) and Energy Analysis Tools with Green Building Certification System to Conceptually Design Sustainable Buildings." *ITcon* 19: 494-519. http://www.itcon.org/2014/29

Jalaei, Farzad, Ahmad Jrade, and Mahtab Nassiri. 2015. "Integrating Decision Support System (DSS) and Building Information Modeling (BIM) To Optimize the Selection of Sustainable Building Components." *Journal of Information Technology in Construction* 20: 399-420. http://www.itcon.org/2015/25

di Giulio, Roberto and Quentin, Costanza and van Nederpelt, Stefan. 2015. "Mapping of energy-related problems and potential optimisation." Tech. rep. D1.3, Streamer consortium.

Mondrup, Thomas Fænø, Jan Karlshøj, and Flemming Vestergaard. 2014. "Building Performance Simulation software for planning of energy efficiency retrofits." *Nordic Symposium on Building Physics.*

Negendahl, K., T. Perkov, and A. Heller. 2014. "Approaching Sentient Building Performance Simulation Systems." *Proceedings of eCAADe 2014.* 49-60.
Negendahl, Kristoffer. 2015. "Building performance simulation in the early design stage: An introduction to integrated dynamic models ." *Automation in Construction* 54: 39-53. doi:http://dx.doi.org/10.1016/j.autcon.2015.03.002.

Negendahl, Kristoffer, and Toke Rammer Nielsen. 2015. "Building energy optimization in the early design stages: A simplified method ." *Energy and Buildings* 105: 88-99. doi:http://dx.doi.org/10.1016/j.enbuild.2015.06.087.

Niknam, Mehrdad, and Saeed Karshenas. 2015. "Sustainable Design of Buildings using Semantic BIM and Semantic Web Services." *Procedia Engineering* (Elsevier BV) 118: 909-917. doi:10.1016/j.proeng.2015.08.530.

O'Donnell, James T. and Maile, Tobias and Rose, Cody and et al. 2013. "Transforming BIM to BEM: Generation of Building Geometry for the NASA Ames Sustainability Base BIM." Tech. rep., Lawrence Berkely National Laboratory. doi:10.2172/1168736.

6.5.1 Tools:

CENtool:
Diva-for-rhino, http://www.solemma.net/
EnergyPlus: https://energyplus.net/
Sefeira: http://sefaira.com/
Simergy: http://www.digitalalchemypro.com/html/products/DAProducts_Simergy.html

6.6 Design Decision Support

Kaplan, Robert S., and David P. Norton. 1996. *The balanced scorecard: translating strategy into action.* Harvard Business Press.

6.6.1 Tools

RE Suite: http://www.demobv.nl/whatwedo_software_RESuite

6.7 Typlogies

Netherlands Board for Healthcare Institutions. 2007. *Building Differentiation of Hospitals – Layers approach.* Utrecht: Netherlands Board for Healthcare Institutions (NHBI).

Streamer research project on hospital districts: http://www.streamer-project.eu/

Università degli Studi di FERRARA

MODULO DI PRESENTAZIONE DELLA TESI DI DOTTORATO DI RICERCA (da allegare alla domanda di Esame Finale e da consegnare all'ufficio Dottorato e Alta Formazione entro i termini previsti)

Corso di Dottorato di Ricerca in ARCHITETTURA E PIANIFICAZIONE URBANA (D.M. 45/2013) - ciclo 841-29

N. di Matricola 118462

Dissertazione di Esame Finale di Dottorato di Ricerca nella II SESSIONE CICLO 29 dell'anno solare 2016

presentata da LANG THORSTEN

NATO A Arnhem - PAESI BASSI - IL 14/09/1975

Titolo della Tesi: Semantic BIM Design Methodology for Energy Efficient Building.
Lingua: INGLESE
Settore scientifico disciplinare: ICAR/12
Tutore: DI GIULIO ROBERTO
Tutore: BORG RUBEN PAUL
Carattere della Tesi: Elaborato Scritto

Firma del Tutore..

Firma dell' eventuale Cotutore: ....

FERRARA, li 31/10/2017 Firma del Dottorando..

Richiedo il rilascio del diploma di Dottorato di Ricerca.

FERRARA, li 31/10/2017 Firma del Dottorando..

Thorsten Lang

Semantic BIM Design Methodology for Energy Efficient Building

International Doctorate in Architecture and Urban Planning (IDAUP)
International Consortium Agreement between University of Ferrara Department of Architecture (DA) and Polis University of Tirana (Albania) and with Associate members 2014 (teaching agreement) University of Malta/Faculty for the Built Environment

Thorsten Lang