EMBODIED OPTIMISATION TOOL

FOR LOW-RISE OFFICE BUILDINGS IN STEEL

by

S. Koonath Surendran

in partial fulfillment of the requirements for the degree of

Master of Science
in Civil Engineering

at the Delft University of Technology,
Faculty of Civil Engineering and Geosciences,
BEMNext Laboratory

In collaboration with
Arup, Bouwen Met Staal & Tata Steel

Student number:	4051084	
Chairman:	Prof. dr. ir. J. G. Rots,	TU Delft
Thesis committee:	Dr. J. L. Coenders,	TU Delft, White Lioness technologies
	Ir. J. W. Welleman,	TU Delft
	Ir. J. P. den Hollander,	Bouwen met Staal
	Ir. B. H. Bonnema,	Tata Steel
	Ir. A. Rolvink,	TU Delft, White Lioness technologies

29 August 2014

TU Delft Delft University of Technology

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank everyone who played a pivotal role in helping me with this project.

First and foremost, I am very grateful to *each and every member of my thesis committee*, for all your ideas, expertise and feedback while working on this project, and more importantly, I thank you for your patience with me. In particular, I am much obliged to *Jan-Pieter* for coming up with this idea in the first place, and to *Jeroen*, for giving me the opportunity and ample guidance to develop this idea into a full-fledged project.

I would also like to thank my *colleagues at Arup, Amsterdam* for keeping my days there intersting, be it with helpful information or with jokes. *My friends - especially Arjun, Manu, Pandit, Sharmishta, Sumeet* and *my housemates*, thank you for all your love and support. A special shout out to the *Roosma* family; I could not have gotten this far, without your encouragement and help. Same goes to *Mr. Karel Karsen, Mr. John Stals, Mr. Lambert Houben* and *Mr. Marco Poot*.

And finally, I would like to thank *my parents*, for everything.

S. Koonath Surendran
Delft, June 2014

Committee Members

- Prof. dr. ir. J. G. Rots (Chairman)
  Delft Univeristy of Technology
  Faculty of Civil Engineering and Geosciences
  Section of Structural Mechanics
  Roof 6.71, Stevinweg 1, 2628 CN Delft
  +31 (0)15 278 3799
  J.G.Rots@tudelft.nl
- Dr. ir. J. L. Coenders (Daily Supervisor)
  Delft Univeristy of Technology
  Faculty of Civil Engineering and Geosciences
  BEMNext Laboratory
  Room: 6.66, Stevinweg 1, 2628 CN Delft

  White Lioness technologies
  Van Diemenstraat 118, 1013 CN Amsterdam
  +31 (0)20 737 1997
  jeroencoenders@white-lioness.com
- Ir. J. W. Welleman
  Delft University of Technology
  Faculty of Civil Engineering and Geosciences
  Section of Structural Mechanics
  Room: HG 6.65, Stevinweg 1, 2628 CN Delft
  +31 (0)15 278 4856
  J.W.Welleman@tudelft.nl
- Ir. J. P. den Hollander
  Bouwen met Staal
  Postbus 190, 2700 AD Zoetermeer
  +31 (0)79 353 1277
  janpieter@bouwenmetstaal.nl

- Ir. B. H. Bonnema
  Tata Steel Construction Center
  2H-14, PO Box 10.000, 1970 CA Ijmuiden
  +31 (0)25 149 5931
  bauke.hoekstrabonnema@tatasteel.com

- Ir. A. Rolvink (Daily Supervisor at Arup)
  Delft University of Technology
  Faculty of Civil Engineering and Geosciences
  BEMNext Laboratory
  Room 6.66, Stevinweg 1, 2628 CN Delft

  White Lioness technologies
  Van Diemenstraat 118, 1013 CN Amsterdam
  +31 (0)20 737 1997
  ankerolvink@white-lioness.com

SUMMARY

About 40-50% of global raw materials are currently being used in the building industry to manufacture building products and related items. At the same time, they are also responsible for 40-45% of total worldwide anthropogenic carbon dioxide emissions (Huovila 2007). Within the industry, both these problems have led to the development of legislative policies, regulations and targets to limit material and energy use in buildings.

Usually, most of the embodied energy and some of the operating energy of a building lies within the structure (Target Zero 2011). This makes it possible to minimize the energy consumption of a building by varying its structural design. Another possibility to lower the annual energy consumption of a building is by increasing its adaptability. However, some aspects of improving adaptability of a building may lead to an increase in its embodied energy consumption. Similar situation applies to the operating energy and embodied energy relationship. Hence, there is potential to apply computational methods to optimise a building design, to have minimal embodied and operating energy consumption while aiming for maximum adaptability.

This thesis investigates the development and application of a computational tool that optimises the conceptual stage design of a building to have minimum embodied energy and some aspects of the operating energy, depending on the adaptability required. For this purpose, a parametric computational framework for sustainable building design was developed and implemented by the tool. The working prototype of the tool focuses on low-rise rectangular grid office buildings in steel. The various competing objectives are optimised by applying multi-objective optimisation techniques.

During the design and development stage of this project, several functional requirements (features) of the tool were identified and incorporated, via a test-driven development methodology. These requirements are broadly classified as:

- Design a building;
- Calculate embodied energy consumption;
- Calculate operating energy consumption;
- Optimise design or product to achieve (one or more) sustainability objectives;
- Export design to operate with (one or more) third-party tools.

Test cases were prepared for each feature, in order to render the tool as functional. Also, the results from each test case were validated to ensure that the tool can be trusted. Aside from the functional requirements, it was also aimed to provide non-functional attributes to the tool such as adaptability, interoperability, reliability, etc. Finally, conclusions are drawn on the framework and the tool, and its limitations and possible future developments are discussed. The main conclusions of this project are summarised below:

- The *Embodied Optmisation Tool* is able to aid in studying the influence of embodied energy, operating energy and adaptability on the design of a building, from within a single tool, thereby aiming to provide answers relating to sustainability, as early as in the conceptual design stage of a building.
- Since the tool aids in decision making from the conceptual stage of the building until its entire lifecycle, it can be concluded that both the framework and the tool are in accordance with the Building Information Modeling (BIM) goals.
- In the area of energy efficient building design, the tool is able to cater to the demands of the key players in the building industry - architects, engineers, building material and products manufacturers, as well as researchers;
- Both the framework and the tool follows a modular approach to building design, such that new features can be easily added or old features can be modified or replaced without affecting the rest of the functionalities, resulting in a flexible tool with a lot of development potential.

CONTENTS

1

INTRODUCTION

Non-sustainable practices (such as deforestation, mineral resource exploitation, industrial pollution, etc), amongst others, has caused our planet to suffer from various socio-economic-environmental issues such as poverty, inequality, uneven development, population growth, depletion of various raw materials (minerals, fossil fuels, biodiversity, land and water). It has also lead to an increase in emission of harmful substances like greenhouse gases, smog, ozone depleting substances, and other toxic components into the atmosphere. When left unchecked, these issues can cause hindrance to human progress and survival (Jonkers 2011).

Our Common Future, more commonly known as the Brundtland Report, was the first official step taken by the world towards the goal of sustainable development. *Sustainable development* refers to the management of various environmental resources for the sustenance of present and future generations (Lebel & Kane 1987). The Brundtland Report aims at developing long term strategies that can lead to an era of economic growth that is socially and environmentally sustainable. It addresses and appeals to people from all walks of life - governments, private enterprises, non-governmental organizations, educational institutions, scientific communities, and individuals; for their participation to bring about the necessary change. More than two decades since then, sustainable development has become a widely recognized term and is a common topic of discussion for all.

This chapter introduces the research problem, i.e., the problem definition of this project. This is followed by an outline of the project report.

1.1. RESEARCH PROBLEM

Mineral resources are depleting The building industry accounts for consumption of about 40-50% of global raw materials to manufacture building raw materials and related items (Huovila 2007). Based on the average annual rate of mineral consumption in the world, the life expectancies of *identified* economic world reserves for most common mineral commodities show that these reserves will run out in about 200 years (some minerals sooner than others). In Europe, these reserves are already almost depleted. Meanwhile, the earth's resource base contains large quantities of minerals that are yet undiscovered, or discovered but uneconomical to extract minerals from. Economists believe that with new technology, market incentives, public policies, material substitution and recycling; the current situation can be mitigated (International Institute for Environment & Development 2002). Many legislative policies are currently under development for building industry to limit this material use and related carbon emissions. The European Union has created the CEN/TC 350 for this purpose and the Dutch Green Building Council is to introduce new regulations on usage of building materials.

Importance of embodied energy has increased Keeping with the goals of Brundtland Report, several policies (at local, national and international level) have been formulated to encourage energy-efficient buildings. They include legislative instruments, economic incentives, technology transfer programmes, and information and education campaigns. Embodied energy comes up to only about 10% of the operating energy during the life-time (60 years) of a building (Institution of Structural Engineers 1999). However, owing to the aforementioned building policies and regulations encouraging energy-efficiency, operating energy of buildings have decreased and are still decreasing. This means that the ratio between embodied energy to operating

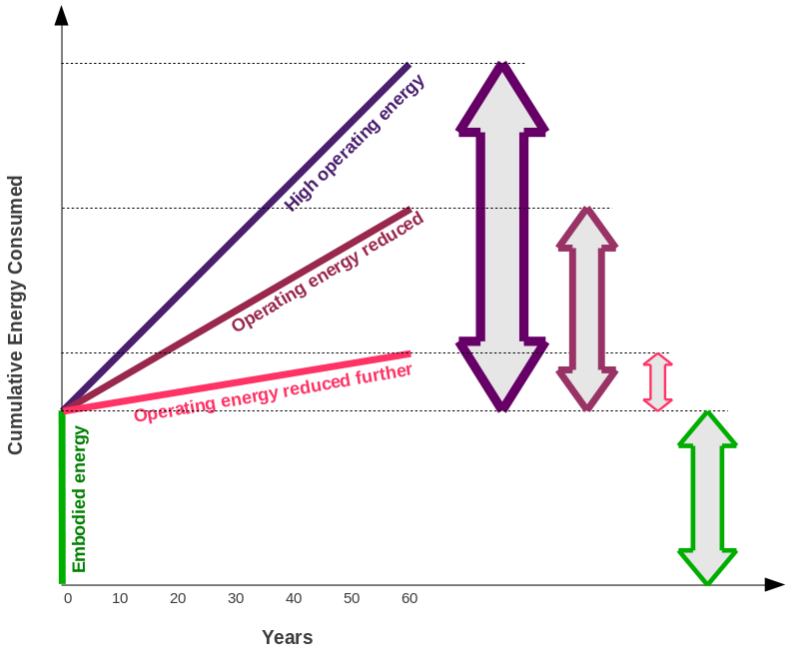

Figure 1.1: An illustration of the increase in importance of embodied energy, as operating energy decreases for a building, based on Yalniz (2008)

energy during the life time of the building has increased, thereby increasing the significance of embodied energy (Figure 1.1) on the total carbon footprint of a building (Yalniz 2008).

Computation can potentially minimize embodied energy consumption in buildings In order to minimise embodied energy of a building, currently the focus is on making the right choice of materials, which also depends on how they are handled after demolition. Structural design can also minimise embodied energy based on the selection and optimisation of the significant structural elements. A study conducted in Arup on an office building found that the load bearing structure contributes on an average 82% of the embodied energy, and that the embodied energy of a structure could be reduced by 20-30% based on the selection and optimisation of the various structural elements (Perkins & Tandler 2005). Another possibility to lower embodied energy consumption of a building over its life-span is by increasing the adaptability of a building. At the same time, some of the aspects of adaptability may also result in an increase in its embodied energy consumption. For example, a replacement of the façade during the life of the building adds to its embodied energy consumption. In a similar manner, an increase in embodied energy can sometimes mean a decrease in the operational energy (Figure 1.2). For example, addition of a new façade with increased thermal insulation or photovoltaic cells. Therefore, it can be seen that there exists a trade-off between the *minimum embodied and operating energy versus maximum adaptability* criteria. Owing to the large amount of variables and calculations involved, it is next to impossible to manually arrive at an optimum solution. However, computational tools opens up the possibility of simultaneously optimising these objectives to arrive at the optimum designs.

1.1.1. PROJECT OBJECTIVE

Based on the aforementioned three points, the main *objective* of this project is to

> **Develop a tool that supports decision making for the conceptual design stage, by assessing and optimising a building design so as to minimise its embodied energy and some aspects of the operational energy, depending on the adaptability required.**

While achieving this objective, it will also be aimed to answer the following questions:

- What parameters define this tool and why?
- What challenges were overcome during the tool development?
- What are the final limitations of the tool? This can be sub-divided into:

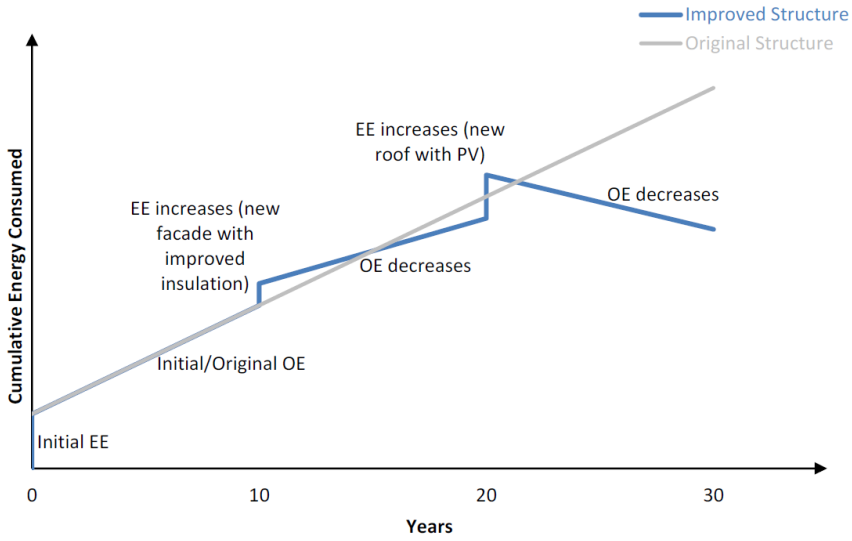

Figure 1.2: Example of energy consumption of an *improved* office building, based on Allwood & Cullen (2012)

  - Risks;
  - Level of control and insight provided;
  - Re-usability/ flexibility/ adaptability/ customisability;
  - Interactivity/ interoperability.

- What are the lines of possible future developments and explorations?

1.1.2. PROJECT SCOPE

The scope of this thesis project provides further clarity regarding the project objective. It has been divided into two categories - *restrictions* imposed on the objective and *assumptions* made in order to achieve this objective.

Restrictions Based on the availability of data for tool development, the following restrictions have been applied:

- Region is Western-Europe;
- Design codes followed are Eurocodes;
- Building material is steel (role of steel in sustainable buildings is explained in Chapter 2);
- Building function is office use;
- Life-cycle analysis aspects considered are energy efficiency and carbon emissions of a building.

Remoy & van der Voordt (2009) states that 13% of office space in the Netherlands are currently vacant due to inability to adapt. In some business areas such as Amsterdam-Zuidoost and Amsterdam-West, this figure is well above 30% (ter Mors 2011). Due to this, design and development of new office buildings need to and are now looking at improving its adaptability. Therefore, it is useful to restrict the building function to office use.

Assumptions The following assumptions have been made in the design of the office building:

- Structure is a low-rise (or medium-rise) braced frame in steel;
- Horizontal sway stability is provided via diagonal wind bracing system in steel;
- Beam-column connections are pinned or hinged connections (therefore, simple construction);
- Façade (including roof) is replaced during the life of the building.

The adaptability rating is based on Tool (2010), therefore, the assumptions made in the same are applicable here.

1.2. THESIS OUTLINE

This thesis report has been divided into three parts, with each part containing several chapters. *Part I* termed *Research* covers the literature review phase of the project. In order to acquire sufficient knowledge to fulfill the project objective, it is necessary to understand in detail the different topics pertaining to the research problem. *Chapter 1* introduces the research problem, the objective and scope of this project. *Chapter 2* covers the most important terms and concepts used in this project. An explanation of the process of multi-objective optimisation together with its role in designing energy efficient buildings is next, in *Chapter 3*.

Once these topics have been covered, the report moves on to *Part II* which consists of the *Design and Development* phase. *Chapter 4* explains in detail, the computational framework developed for the tool, the requirement specifications and the final design of the tool. In *Chapter 5*, the development methodology adopted is explained, followed by the various test cases for validation of the tool and the results obtained.

Part III is the final part of the report, containing the *Discussions, Conclusions and Recommendations* on the project. The achievement of the project objective and the limitations of the tool are discussed in *Chapter 6*. Finally, *Chapter 7* presents the conclusions drawn from this project, which is followed by the recommendations for further studies and development of the tool in *Chapter 8*.

2

ENERGY EFFICIENT BUILDINGS DESIGN

This chapter provides an insight into sustainable or green buildings and related terms and concepts used throughout this project.

2.1. GREEN BUILDINGS

The life of a building can be classified into several stages (Figure 2.1):

- Product stage;
- Construction process stage;
- Use stage -
  - Operation stage;
  - Maintenance stage;
- End-of-life stage.

The *product stage* consists of acquiring raw materials from the earth, transportation of these raw materials to factories and manufacturing building materials and products from them. *Construction process stage*, as the name suggests, refers to the various processes involved in the construction of a building. This includes transportation of the manufactured building products from factories to the building site and their installation procedures.

Once the construction is complete, the building is ready for use. This marks the beginning of the *use stage* of the building, which can be sub-divided into *operation* and *maintenance stages*. A building is said to be in operation when it is under occupancy, during which time there occurs consumption of large amounts of energy and water. After a certain duration of operation, the building undergoes maintenance and repair in order to continue the operation of the building. The *operation* and *maintenance stages* alternates until the building is at the end of its life. In the *end-of-life stage*, the building is demolished and its materials are transported for disposal or recycling or re-use.

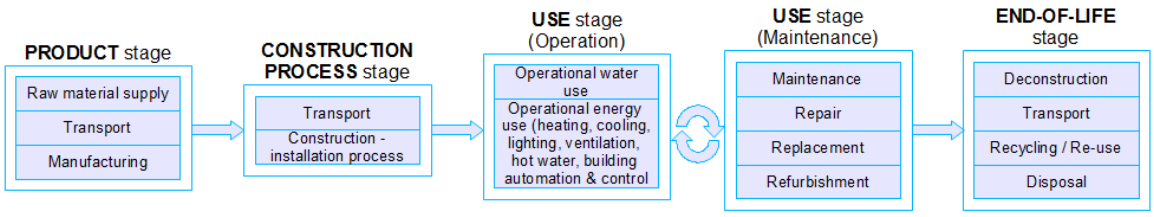

Figure 2.1: Building life-cycle scheme (divided into stages and modules) based on EN 15804

Each of the above-mentioned stages consume energy, raw materials and other natural resources, to varying extents. Non-sustainable practices may be followed in some or all of these stages resulting in wastage, pollution, and environmental degradation, leading to adverse effects on human well-being (Environmental Protection Agency 2014). According to Huovila (2007), buildings account for about 30-40% of the worldwide energy consumption. In Europe, this figure is about 40-45%. Consequently, the building sector is responsible for around 40-45% of total worldwide (and European) anthropogenic CO2 emissions.

The concept of *green buildings* was developed to reduce, mitigate and prevent non-sustainable practices in the building industry. A building may be defined as green, by assessing whether it is environmentally responsible and resource-efficient throughout its life (Environmental Protection Agency 2014).

2.1.1. SUSTAINABLE BUILDING RATING SYSTEMS FOR GREEN BUILDINGS

Several *sustainable building rating systems* have been developed across the world to provide standards and guidelines for developing green buildings. The most well-known international methodologies among these are Building Research Establishment Environmental Assessment Method (BREEAM) in United Kingdom, and Leadership in Energy and Environmental Design (LEED) Green Building Rating System in United States (Wong 2010). Most of these rating systems follow a point-based/check-list approach, by which, points are allotted to buildings for satisfying various green building standards and criteria.

However, to assess the environmental impact of whole buildings, it is necessary to rate them based on their actual environmental performance during their life. Thus the focus has now shifted towards incorporating life-cycle assessment techniques into the certification procedure.

2.1.2. LIFE-CYCLE ASSESSMENT

A *Life-Cycle Assessment* or *Life-Cycle Analysis* (LCA) is performed to determine the environmental impacts of any product during its lifespan. These impacts include resource depletion, energy and water use, greenhouse emissions, waste generation, toxicity, land use, ozone depletion, management, health and well-being, etc. (Milne & Reardon 2014).

International Organization for Standardization (ISO) 14040 and 14044 series provides the standards for the LCA methodology. Many LCA tools implementing these standards are available in the market, either as a stand-alone software (for example, SimaPro) or packaged with other systems (for example, Athena Impact Estimator for Buildings). In the Netherlands, both Greencalc+ and GPR Gebouw have integrated LCA tools into their sustainable building rating systems. Since 2008, Greencalc+ has become a part of the Dutch version of BREEAM (Ministerie van VROM 2014).

Note In this thesis, the focus is on energy efficiency and carbon emissions of a building. Other aspects of LCA have not been considered.

2.1.3. ENERGY CONSUMPTION OF A BUILDING

Energy-efficiency is one of the foremost goals of green buildings. The total energy consumed by a building can be divided into 6 phases (Jones 1998) :

- Embodied energy;
- Grey energy;
- Induced energy;
- Operating energy;
- Demolition energy;
- Recycling energy.

Figure 2.2 illustrates the cumulative energy consumed by a building during its life span, thereby providing an indication of the magnitude of energy consumption in each of the phases. Referring back to Figure 2.1, it can be seen that the the different building life-cycle stages are in accordance with the different phases of energy consumption of a building. So, the product stage consists of embodied energy consumption, construction-process stage consists of grey and induced energy consumption, use stage consists of operation energy consumption, end-of-life stage consists of the energy consumed during demolition and recycling.

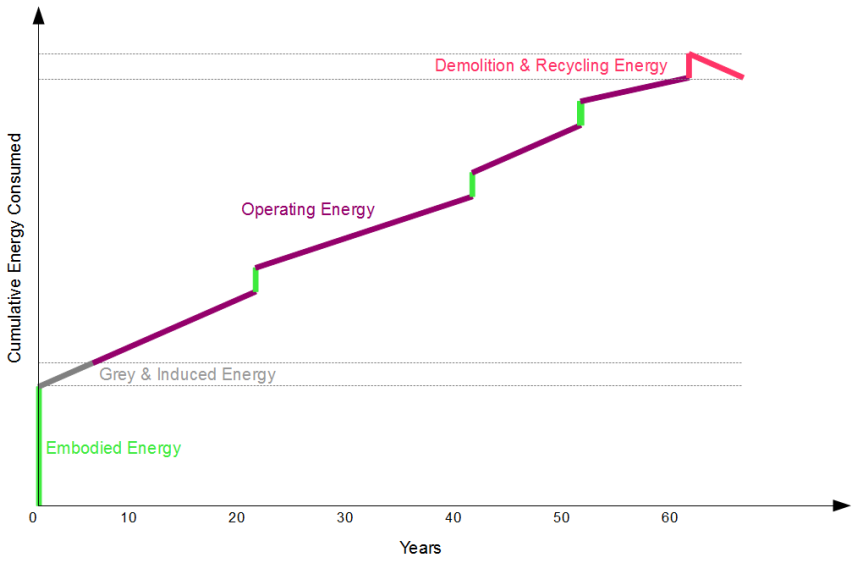

Figure 2.2: Energy consumed in the life of a building

EMBODIED ENERGY

Embodied energy is the total primary energy consumed in mining, transportation (to factory), all the manufacturing processes involved up to the point that any building material (or related components) leaves the factory gate. Hence, it is also known as *cradle-to-gate embodied energy*. It is the second largest phase of energy consumption by a building (as indicated in Figure 2.2. According to Target Zero (2011), structural elements that contribute *significantly* towards the total embodied energy consumption in a building are:

- Foundation;
- Ground floor;
- Remaining floors and roof slabs;
- Load bearing and stability structures (including connections);
- Raised access floors/ceilings;
- External walls and glazing (façade);
- Internal walls;
- Drainage.

Therefore, by choosing the right combination of building materials for each of these structural elements, it is possible to reduce the embodied energy consumption in a building.

Note The more complicated the manufacturing process of a material, the higher is its embodied energy. At the same time, the higher is the environmental pollution, in particular the greenhouse gases. Due to this relation between energy consumed and carbon-dioxide released, another common term that is applied is *embodied carbon-dioxide* or *embodied carbon*. Similar to embodied energy, it can be defined as the amount of carbon dioxide produced over (a part of) the life cycle of a product.

GREY & INDUCED ENERGY

Less data is currently available on *grey and induced energy* consumption phases as they vary widely from place to place, from one construction practice to another, etc. Hence, assumptions are made while calculating these energy values. From Figure 2.2, it can be seen that the quantity of energy consumed during these two stages is very small when compared to embodied energy and operating energy consumption. Due to this reason, most times they are excluded from total energy calculations or they are included as a percentage of the initial (embodied) energy consumption.

OPERATING ENERGY

Operating energy is the energy used to operate a building, i.e., for heating, cooling, lighting, ventilation, water supply, building automation and control, etc., during its use/occupancy. As visible in Figure 2.2, this is the largest portion of energy consumed by a building during its life-cycle. Therefore, it creates the most opportunities for large amounts of energy-saving within a building. The green buildings concept described in the previous chapter has led to the development of several energy-efficiency targets for buildings, such as:

- Low energy buildings;
- Zero energy buildings;
- Energy-plus buildings.

Low energy buildings, as the name suggests consume lower energy for operation, compared to contemporary buildings. *Zero energy buildings* are those which generate sufficient energy annually to compensate for the energy consumed by it, thereby resulting in a net zero energy consumption. A building is rated as an *Energy-plus building*, when it is able to produce more energy in a year than it consumes, thus, leaving a positive energy footprint. These targets are achieved through a combination of techniques which can be broadly classified as:

- Techniques to reduce energy loss, such as providing better insulation, floor heating, sun-shading, LED lighting, etc;
- Techniques to produce energy, such as using solar water heating, photovoltaic, etc;
- Techniques to recycle or reuse energy, such as using ventilation heat recovery systems, geothermal heat pumps, etc.

DEMOLITION & RECYCLING ENERGY

Similar to embodied energy, both *demolition energy and recycling energy* are related to the building materials. For most materials, its service-life ends with demolition, after which the waste is disposed off (usually, dumped in landfills). But sometimes, building material for a new structure is obtained by recycling building material from a demolished structure, which considerably lowers its embodied energy value. A common example for such a material is steel. Such a relation has led to the inclusion of demolition and recycling energy as a part of total embodied energy of a building material. If the material ends with demolition, it is known as *cradle-to-grave embodied energy* and if the material is recycled, it is known as *cradle-to-cradle embodied energy*. Efforts are being made in the building industry to make/develop all building materials to have low cradle-to-cradle embodied energy.

2.2. ADAPTABILITY OR FLEXIBILITY OF A BUILDING

Recycling and reuse of building materials can lower their embodied energy values. But this is not always possible and wastage still occurs at the end of a building's life. Therefore, it is necessary to design buildings that are adaptable for future use (Huovila 2007). *Adaptability* or *Flexibility* refers to designing a structure to accommodate future changes in services, engineering strategy, aesthetic values, architectural trends and the function of the building (Foster & Greeno 2007).

The first step to improve the flexibility or adaptability of a structure lies in identifying the various aspects that influence the life of a building. The next step is to quantify them. According to Tool (2010), these aspects are:

- Stability system;
- Voids flexibility;
- Load bearing capacity;
- Floor height;
- Structural grid size;
- Façade;

- Installations.

Tool (2010) quantifies these aspects by providing a factor range for each of them (Appendix A), using which an estimated service life (ESL) factor can be determined for a building.

$$ESL = Factor_1 * Factor_2 * ... * Factor_n \tag{2.1}$$

This ESL-factor is multiplied with the lifespan of a building to obtain its estimated service life (as opposed to the estimated designed life of a building), using which the annual environmental cost of that building can be calculated.

$$Annual\,environmental\,cost = Annual\,operation\,cost + \frac{Construction\,cost + Demolition\,cost}{Lifespan * ESL} \tag{2.2}$$

From the equation above (2.2), it can be seen that greater the ESL value, lesser is the annual environmental cost, i.e., construction and demolition cost. The construction cost primarily refers to embodied energy consumption.

2.3. STEEL AND SUSTAINABILITY

The building industry has already begun taking steps towards lowering operating energy consumption, hence the focus now shifts towards reducing the second largest energy consumption, i.e., embodied energy consumption of buildings. This is largely dependent on the building materials and the various building products currently available in the market.

Amongst these materials, steel in particular, possess several sustainable properties in comparison to its counterparts (Tata Steel Construction 2014):

- All grades of steel can be repeatedly recycled without undergoing any loss of performance. In UK, the recycling and reuse rate for steel construction products is 94% and around 99% for structural steelwork;
- Steel frames can be easily dismantled and relocated;
- High strength-to-weight ratio of steel leads to smaller foundations and therefore lesser carbon footprint;
- High strength-to-weight ratio of steel leads to longer spans and therefore greater flexibility;
- Modern steel cladding systems comes with high insulation and air-tightness;
- All steel construction products are prefabricated to correct dimensions, leading to high quality, fewer deliveries to site, rapid construction and less waste on site;
- Steel frames can be easily dismantled and relocated.

Tata Steel Construction (2014) found that more than 70% of multi-storey commercial steel buildings in UK are made of steel frames. This knowledge puts steel in a unique position to be able to reduce energy consumption in a building. Steel-makers are, therefore, coming up with new developments such as core panels that allow for fast construction of structural cores for multi-storey buildings, carbon-neutral building envelopes, combining underfloor heating/cooling to composite floor decking, solar panels on roof cladding systems, etc. (Tata Steel Construction 2014).

3

MULTI-OBJECTIVE OPTIMISATION

In order to develop a computational tool that can optimise the structural design of a building, it is first necessary to understand the procedure of optimisation in detail. The aim of this chapter is to introduce this concept and its application in this thesis project.

3.1. OPTIMISATION

Optimisation refers to the action of arriving at an optimum solution for a given problem. To define optimisation mathematically, it is necessary to familiarise with the following terms:

- *Decision variable*, say '*x*'. Variable, as the name suggests, is a factor that can vary or change its attributes or values. The word decision is added to the term because the outcome of the problem is decided by changing the different values of the variable. An optimisation problem can have multiple decision variables.

- *Constraint*, say '*g(x)*'. Generally, optimisation problems have several restrictions or constraints imposed on them. This may be due to certain environmental characteristics or based on the availability of resources (physical limitations, time restrictions, etc.) Only those solutions that satisfy these constraints are deemed acceptable (Coello et al. 2002). These constraints can be inequality constraints (<. <=, >, >=) or equality constraints (=), describing the relationship between decision variables and constants or parameters in the problem.

- *Objective function*, say '*y = f(x)*'. Objective function is a function that is dependent on the decision variable such that the value of this function is to be optimised (maximised or minimised).

Using the above terms, optimisation can be defined as

$$\begin{array}{llll} maximise & y = f(x), & where & x = x_1, ..., x_n \\ subject\ to & g_i(x) <= 0 & where & i = 1, ..., m \end{array} \tag{3.1}$$

Mathematically speaking, it can be said that a (structural) design model consists of a combination of several design variables, fixed parameters, and constraints placed on these variables. The process of (structural) design involves alternating rounds of fine-tuning the values of these design variables and testing the resultant design models, until a suitable or *optimum design* is arrived at. In other words, the process of structural design is nothing but an optimisation procedure.

Therefore, by treating the process of structural design as an optimisation procedure, it is possible to arrive at the most optimum design solutions for sustainability. Owing to the importance of decisions made in the early stages of design, it is especially useful to develop computational tools that aid in decision-making from the conceptual design stage (Rolvink 2010).

3.2. MULTI-OBJECTIVE OPTIMISATION

Most real-world problems (for example, design of a building) have multiple decision variables and have to simultaneously optimise various (often competing) objectives to arrive at the best solution(s). Such an optimisation is known as *multi-objective optimisation* or *multi-criteria optimisation*. Mathematically, the goal of multi-objective optimisation is written in the following format:

$$\begin{array}{llll} maximise & y = (f_1(x), f_2(x), ..., f_k(x)) & where & x = x_1, ..., x_n \\ subject\,to & g_i(x) <= 0 & where & i = 1, ..., m \end{array} \tag{3.2}$$

Single objective optimisation problems lead to a single optimal or best solution. With multi-objective optmisation problems, this is not possible, as there occurs different solutions that are the best solutions based on the different objectives. Instead, a set of alternative trade-offs called *Pareto-optimal solutions* are arrived at, which are the optimum solutions within that search space (Zitzler 1999).

3.2.1. CLASSICAL METHODS

The classical or traditional methods of multi-objective optimisation consists of combining the multiple objective functions into a single aggregate objective function, using which the Pareto-optimal solution set is generated. For example, in the *Weighting Method*, the multi-objective optimization problem is converted into a single objective optimization problem by providing weights to each objective and linearly combining the different objectives such that the sum of the weights = 1.

Another example is the *Constraint Method* where a multi-objective optimization problem with k objectives is converted into a single objective optimization problem by transforming k - *1* objectives into constraints so that only one objective function remains.

Such traditional approaches were found to be lacking in regards to generating the *complete* Pareto-optimal solution set (Zitzler 1999). Also, depending on the situation, several independent optimisation runs are necessary to obtain the Pareto-optimal set, which leads to greater computation load. Hence a need arose for alternatives to the classical methods, that can overcome such shortcomings. One such alternative is the approch of evolutionary algorithms.

3.2.2. EVOLUTIONARY APPROACH TO MULTI-OBJECTIVE OPTIMISATION

Evolutionary algorithms can handle large search spaces and perform multiple alternative trade-offs in a single optimisation run. They are also highly flexible and can be adapted to suit any problem. Genetic algorithms and evolutionary programming are two examples of evolutionary approach to multi-objective optimisation. Genetic algorithms imitates the process of natural evolution - survival of the fittest. This approach can be explained in the following steps:

- A set of solution candidates is maintained. The candidates are called individuals and the set is called population;
- Generate an initial population (random or predefined);
  - The individuals in the population are evaluated on their quality (fitness), such that the weak-individuals are eliminated and the healthy-individuals are selected for reproduction;
  - Reproduction consist of recombination of the parents (cross-over) and mutation (of individuals) to produce off-springs;
  - The parents are then replaced by the fitter off-springs, to form a new generation.
- Repeat step (2) for the new generation, until convergence or the predefined maximum number of generations are met.
- The final population contains the fittest off-springs and is the Pareto-optimal solution set.

Several different evolutionary algorithms are available today, each different in its approach to fitness assignment, or population generation, etc. While evolutionary approach has its advantages, there is no clarity on:

- Which algorithm is superior to others;
- Which algorithm suits which problem better, etc.

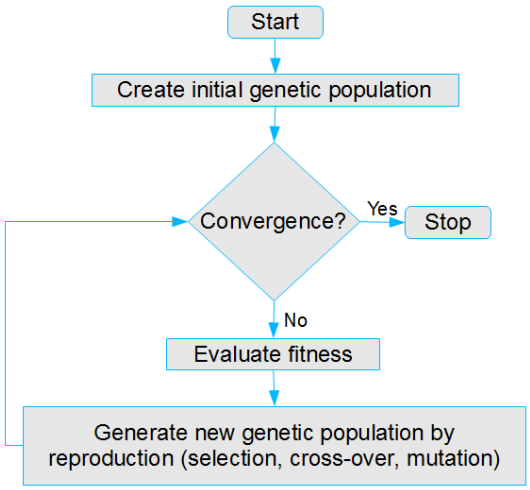

Figure 3.1: Flowchart for Evolutionary Algorithm

3.3. APPLYING MULTI-OBJECTIVE OPTIMISATION IN BUILDING DESIGN

The goal to design energy efficient buildings creates the opportunity to apply optimisation to the structural design process. One such possibility is to organize the design process in the form of a genetic algorithm, such that the most energy efficient design can be obtained with the aid of a computer. The example below demonstrates this idea.

On the left half of Figure 3.2, there are 4 design variables. For the optimisation algorithm to work with these variables, each of them are provided with binary codes, which acts as *genes*. The right-hand side of this figure shows the different possible combinations of these variables (genes) to make *chromosomes*. Some of these combinations may result in weak or bad designs, some of them will be the best.

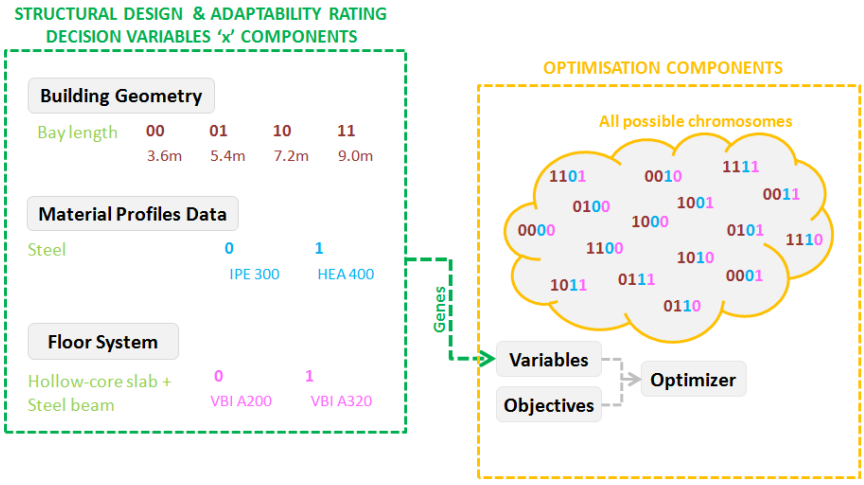

Figure 3.2: MOP in Building Design: All possible combinations of design variables

Figure 3.3 shows the initial population consisting of randomly selected chromosomes. These combination of design variables go through the structural calculations. Based on the building code checks, only those combinations are passed after calculations which result in a feasible structure. At the bottom of the figure, the passed chromosomes go into the energy calculator, which determines the energy consumption of the building for these chromosomes. Based on their energy values, each chromosome is provided with a fitness value, such that higher the energy consumption, lower the fitness (or vice-versa).

In theory, one can pass all possible combinations of variables through the design calculations, obtain the passed combinations and determining their energy consumptions. With this information, the most fit combination can be identified as the final design. However, due to the large number of design variables that exist in an actual design, there are countless possible combinations. To save on the time, cost and computing

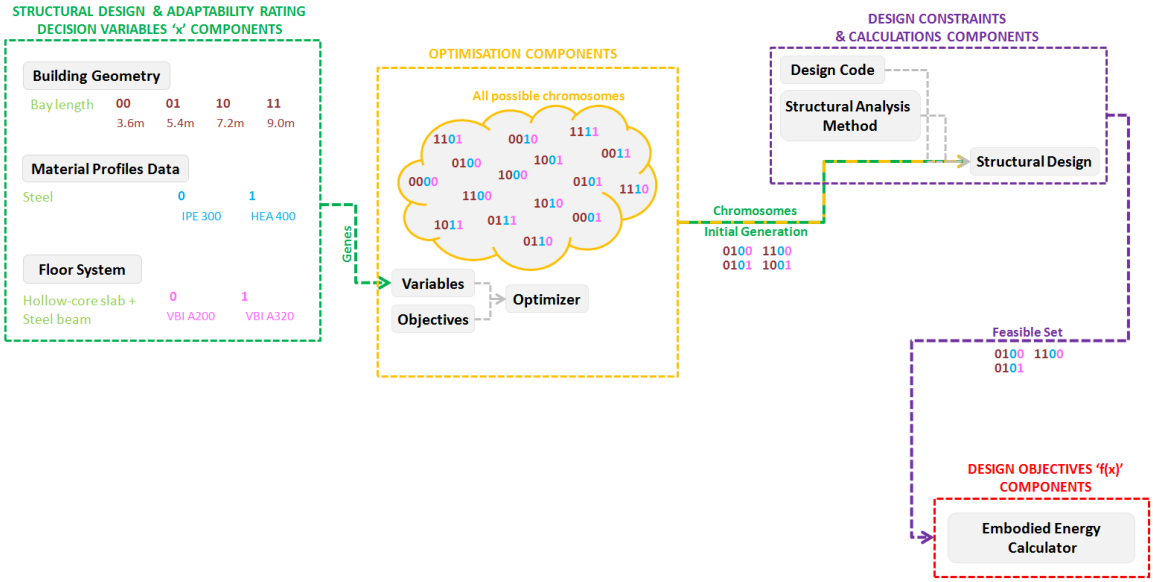

Figure 3.3: MOP in Building Design: Initial population, calculations and assigning fitness to each value

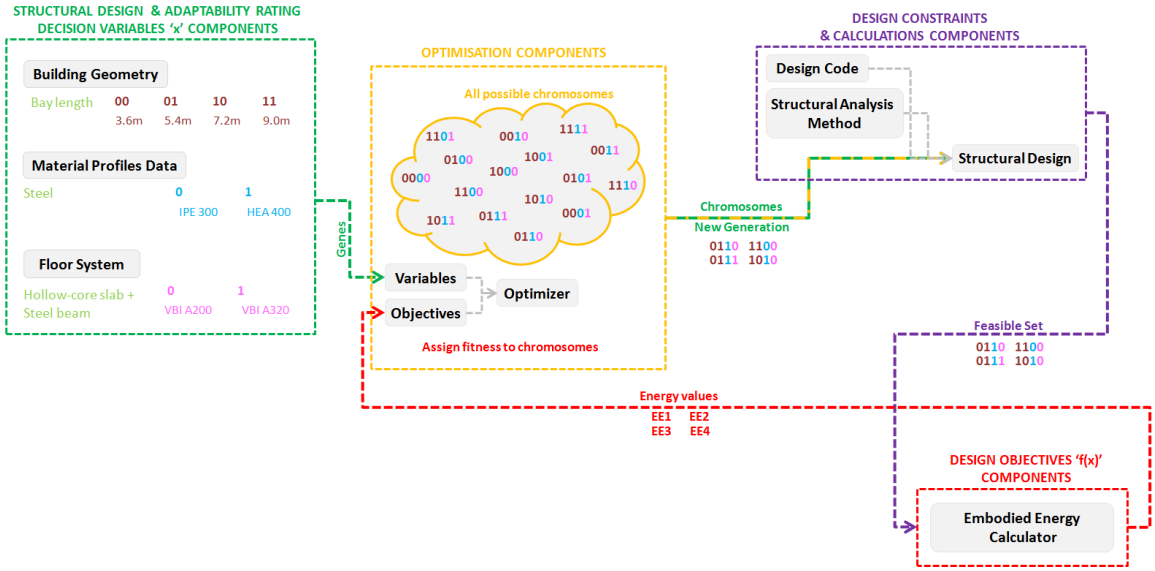

Figure 3.4: MOP in Building Design: New generation, design and fitness

power involved in solving this, it becomes necessary to smartly arrive at the right design without applying brute force methods. Hence, the genetic algorithm.

Figure 3.4 obtains a new generation of chromosomes. This new population is obtained by *selecting* the combinations that are most fit, eliminating the weakest combinations and obtaining new combinations via *reproduction* of the members of previous generation. As visible in the figure, this is done by recombination of the fit parents - *cross-over* and by *mutation* of combinations from the previous generation.

With this new population, the design calculations are performed and the energy consumption of the passed combinations are obtained. Once again, they are provided with fitness values. This process continues and with each generation, the fitness of the population increase until only the fittest combinations remain, or in other words, convergence is achieved.

By generalizing this idea, a computational framework for sustainable building design can be developed. The *Embodied Optimisation Tool* is a software tool built by implementing this framework. This is explained

in the following chapters, which marks the design and development phase of this project.

4

Design of Embodied Optimisation Tool

The research phase of this project concluded with the idea to develop a computational framework around the process of structural design, to obtain an optimum design model. This chapter marks the beginning of the tool development. The first section of this chapter introduces the computation framework, which is then implemented by the *Embodied Optimisation Tool*. This is achieved via the software design and development process.

4.1. A Computational Framework for Sustainable Building Design

In order to develop the framework, the objective of this project is divided into different functional parameters:

- *Design variables* such as structural design variables, building services design variables and adaptability rating variables;
- *Design methods* for structural analysis and design;
- *Objectives functions* such as embodied energy, operating energy, adaptability;
- *Optimisation algorithm*, such as any suitable genetic algorithm.

Each of these parameters are provided with their own interface, and on assigning associations between them, a parametric sustainability framework is created as shown in Figure 4.3.3. Each parametric interface is to have its own primary attributes which is inherited by all of its *children*. This ensures that the framework can easily incorporate new or improved features when necessary, without breaking the functionality of the rest of the framework. The framework by itself is an idea, the next step is to implement it into a software tool.

Note Since the objectives functions in this project are related to sustainability, the framework has been termed as parametric sustainability framework. However, it is also possible to add sustainability independent objectives such as cost-optimisation into the framework if desired.

Just like with any other design process, it is necessary to evolve a good software design at the beginning of any software development. The Institute of Electrical and Electronics Engineers (IEEE) has developed standards and guidelines to aid in this process. The format followed in the following sections is roughly based on the software requirements specification document and software design document used in professional software development.

4.2. Requirements Specification

Development of any product stems from a need from the users. In order to ensure that the final product fully satisfies its user requirements, it is necessary to list out and plan in detail, all the desired features. This section describes the various features or requirements specification for the *Embodied Optimisation Tool*. These features are derived from the project objectives provided in Chapter 1.

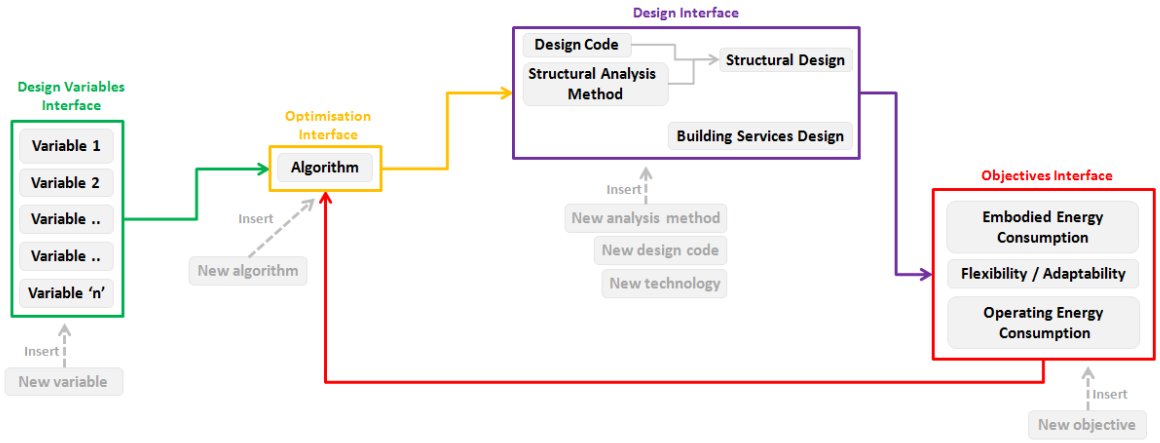

Figure 4.1: A computational framework for sustainability

4.2.1. OVERALL DESCRIPTION

The requirement specifications of the *Embodied Optimisation Tool* consists of:

- User classes;
- Main features;
- Operating environment;
- Dependencies (Appendix B).

USER CLASSES

User classes, simply put, are the different kinds of people who can be expected to use this tool. Based on technical expertise and the different functions in the tool that they may use, the user classes for this tool have been categorised into:

- Designers (architects and engineers);
- Building material/product manufacturers;
- Researchers.

Brief explanation of each of the user classes and on how the features of the tool are applicable to each of them, is provided later in this chapter.

PRODUCT FEATURES

The tool is expected to perform the following tasks for a building:

- Design a building;
- Calculate embodied energy consumption;
- Calculate operating energy consumption;
- Optimise design or product to achieve (one or more) sustainability objectives (minimum energy consumption, maximum adaptability, etc.);
- Export design to operate with third-party tools (Geometry Gym, Karamba, etc.).

These features are described in detail, later in this chapter.

OPERATING ENVIRONMENT

The embodied optimisation tool is built to run on 3D modeling tool Rhinoceros (Version 5.0) via its plug-in Grasshopper (Build 0.9.0075) and on their respective newer releases. Therefore, the system requirements of Rhinoceros are applicable here. Details on Rhinoceros and Grasshopper have been provided in Appendix B.

4.2.2. USER CLASSES & CHARACTERISTICS

Three main user classes (UC) were identified for this tool. They are:

UC-1: Designers (architects and engineers) Nowadays, it is practically impossible to construct a building without the involvement of architects and engineers. They are collectively responsible for evolving a good building design that not only meets the client's requirements, but is also sustainable. Not all information is available to the architects and engineers in the beginning stages of design. With the availability of new information, the initial design undergoes several revisions until the final design is obtained. A vast number of interrelated factors influence the life of a building. Identifying these factors, incorporating them into the design and optimising their values to obtain a sustainable and economic design is a complicated, repetitive and time consuming procedure. The *Embodied Optimisation Tool* can directly aid the designers in simplifying their work; thereby saving time, energy and cost.

UC-2: Building material/product manufacturers In the building industry there is a huge market for different kinds of building materials and products. In order to remain competitive in this market, the manufacturers are always aiming to make products that provide more value for cost. One of the ways to increase the value of a building product is to improve its green building properties, by which, it is necessary to ensure that the product is not only sustainable by itself, but it also helps the building to remain sustainable during its life-cycle. Using the *Embodied Optimisation Tool*, the manufacturer can study the impact of their products on buildings over entire life-cycles. This provides more control in identifying and improving properties of the building products.

UC-3: Researchers The concept of sustainable development is still relatively new, researchers everywhere are exploring the various aspects of sustainability within their respective fields. The building industry is no exception in this matter. The *Embodied Optimisation Tool* provides researchers with a quick and easy option to study some of the environmental impacts of a building over its life-cycle.

4.2.3. SYSTEM FEATURES (FUNCTIONAL REQUIREMENTS)

The main features of this tool were listed earlier in this chapter. They are now described in detail. Each feature is provided with a short description, some use-case situations, followed by the functional requirements that make up the feature.

Each use-case situation is aided with *Use Case Diagrams*, which are diagrams used in software development to describe the functionalities of a software in a pictoral manner. A couple of notes on the use case diagrams:

- *«include»* refers to sub-tasks of the main task
- *«extends»* refers to specialised tasks that can be generalised by the main task.

DESIGN A BUILDING

The tool performs structural analysis and design calculations for a building.

Use Case

USE-1 During the conceptual design stage for a building project, UC-1 uses the tool to generate a quick design model of the building and study the results (Figure 4.2). Alternatively, UC-3 uses the tool for similar purposes.

Functional Requirements

REQ-01 Accept initial design data from the user. The design data is made up of, but not limited to, general information about the project, the building geometry data, the different loads (and load factors) acting on the structure, the structural analysis method for calculating forces in the structure due to these loads, the primary building material, and the design code to be used.

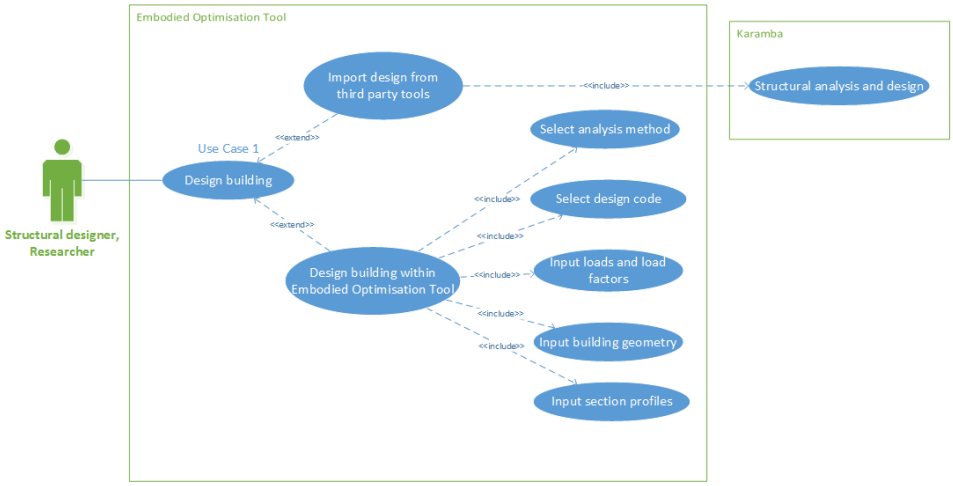

Figure 4.2: Use Case 1 - Design a building

REQ-02 Allow user to input the design data in different formats. General information, building geometry and loads are to be entered manually by the user. Structural analysis method, primary building material and design code are selected by the user from the different options provided in the tool.

REQ-03 The primary building material's strength properties and cross-section profile data should be available as databases within the tool or as external files (Text *.txt*, comma separated value *.csv*, spreadsheet *.xls* or *.ods*, etc.) accessible to the tool. Based on the building material chosen by the user, the tool will access this data for the design.

REQ-04 The analysis methods and design code checks should be hard-coded into the tool. Depending on the user's choice, the tool applies the relevant calculations to the design.

REQ-05 Generate the building design using the input design data.

REQ-06 Provide user with reports on design and analysis calculations.

REQ-07 Provide user with 3D information of the design.

As already mentioned in the project scope, the tool is limited to designing a rectangular grid building via simple braced frame analysis and using the steel Eurocode. However, this is not always the case when it comes to the real world, hence becomes necessary that the tool is able to easily provide option to replace the analysis and design methods used. This adds the following functional requirement:

REQ-08 Taking into consideration of the design limitations of the tool, provide user with the option to override the default (hard-coded) analysis methods, building material data and design code available in the tool with at least one third-party tool.

CALCULATE EMBODIED ENERGY CONSUMPTION

The tool calculates the embodied energy consumption of a building over its life-cycle. The estimated-service-life of the building is calculated to incorporate life-cycle into the calculations.

Use Case

USE-2 In order to achieve a high rating in the sustainable building rating systems, a building needs to have low embodied energy consumption (Figure 4.3). UC-1 or UC-3 uses the tool to determine the embodied energy consumption of the building.

Functional Requirements

REQ-09 Allow user to enter multiple types of input from which embodied energy consumption can be calculated. Possible inputs are quantity of material used, location, structural design data for the building, floor system data, façade system data and life of the building.

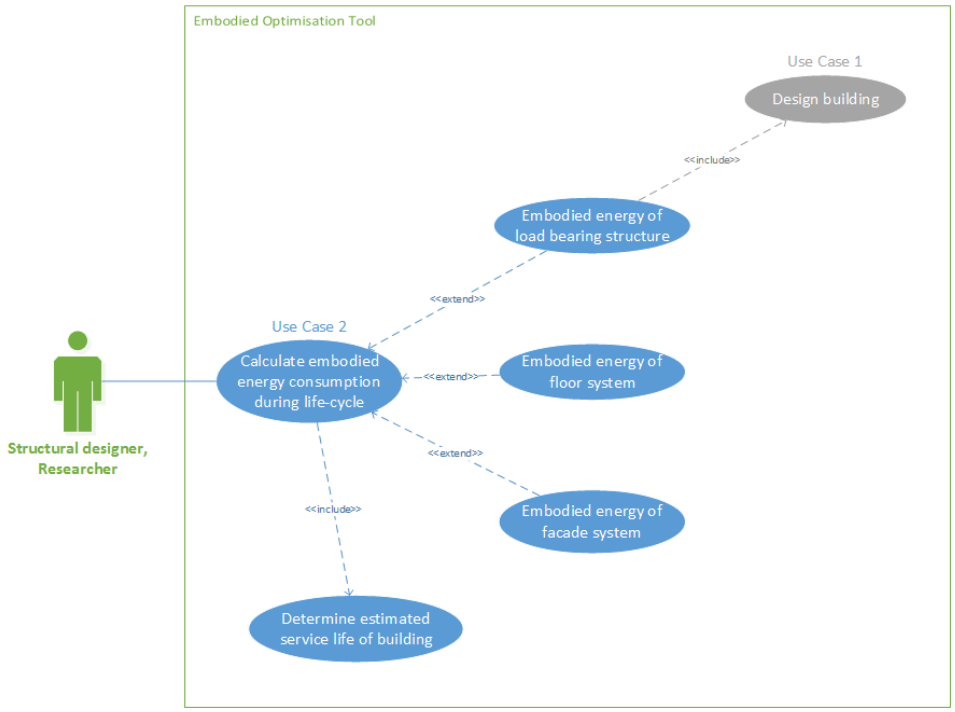

Figure 4.3: Use Case 2 - Calculate embodied energy consumption

REQ-10 Embodied energy values for different building material should be available as databases within the tool or as external files accessible to the tool. Depending on the input location and building materials used, the tool uses the appropriate values for calculation.

REQ-11 Allow user to input the data in different formats. Structural design data is manually entered. Other data such as location and function of building can be choose from a list of given options, for which the embodied energy data is available in the tool.

REQ-12 Provide user with a complex embodied energy calculator. The user inputs the structural design data, from which, the tool generates the structural design for the building (USE-1). From this, the quantity of materials used can be determined.

REQ-13 Allow user to estimate percentage of base materials used in building products not designed by Embodied Optimisation Tool, like foundation, floor systems and input these values into the embodied energy calculator.

REQ-14 The tool should calculate the embodied energy consumption over the life-cycle of the building, hence, not just the initial embodied energy.

REQ-15 Accept flexibility/adaptability aspects data from the user. This data is made up of, but not limited to, accessibility and positioning of installations, area of voids in the floor, imposed floor load, excess vertical and horizontal clearance, grid size of façade, life of the building.

REQ-16 Calculate the estimated service life of the building from the input data.

REQ-17 Provide user with the option to override the estimated service life determined by the tool.

CALCULATE OPERATING ENERGY CONSUMPTION

The tool calculates the operating energy consumption of a building over its life-cycle. The estimated-service-life of the building is calculated to incorporate life-cycle into the calculations.

Use Case

USE-3 Similar to USE-2, the tool can be used by UC-1 or UC-3 to study the operating energy consumption of a building (Figure 4.4).

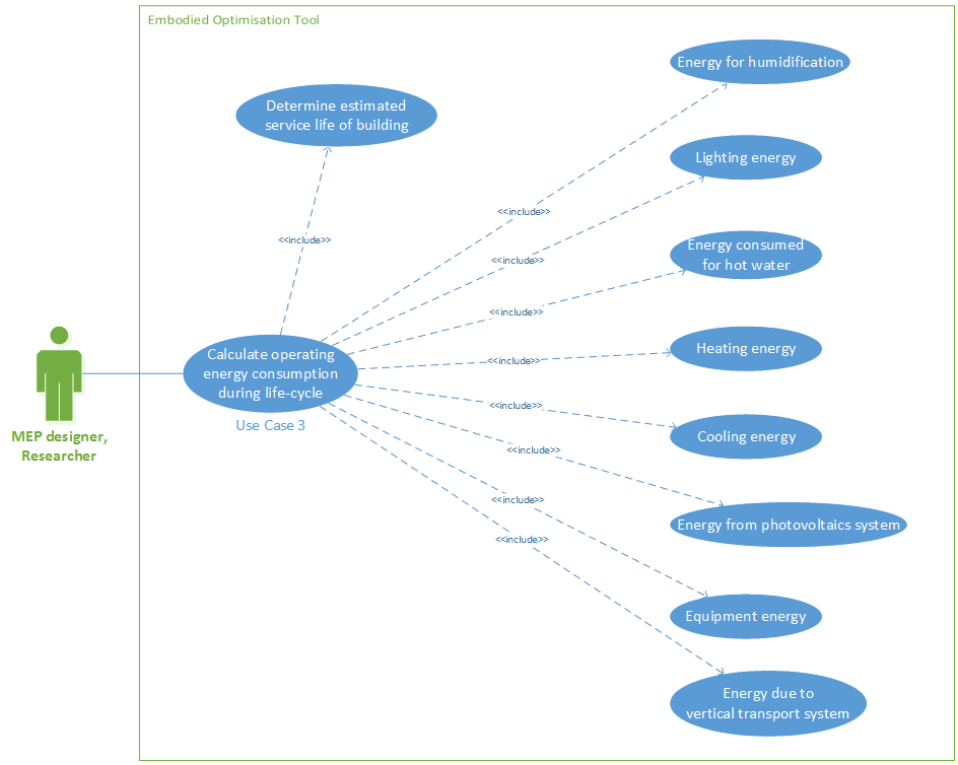

Figure 4.4: Use Case 3 - Calculate operating energy consumption

Functional Requirements

REQ-18 Accept initial operating energy data from the user. This data is made up of, but not limited to, the building geometry data, occupancy data, temperature and humidity requirements, weather data for the building location, façade (and roof) data, electrical equipment data, natural and artificial lighting requirements, information on HVAC (heating, ventilation and air-conditioning) system to be provided, information on the water supply system to be provided, information on the vertical transport system (if any), photovoltaic data (if provided in the building) and information on the different types of energy sources available.

REQ-19 Allow user to input the data in different formats - manual entering and choosing from a list of given options.

REQ-20 Weather data should be available as databases within the tool or as external files accessible to the tool. Based on the location chosen by the user, the tool will access the data for the design.

REQ-21 Calculate the annual operating energy consumption of the building from the input data, as this value determines whether the building achieves targets like low-energy, zero-energy or energy-plus buildings. Energy calculation method should be hardcoded into the tool. Depending on the user's inputs, the tool applies the relevant calculations on the design.

REQ-22 The tool should calculate the operating energy consumption over the life-cycle of the building. For this purpose, the estimated service life is required, hence REQ-15 to REQ-17 are also applicable here.

OPTIMISE DESIGN OR PRODUCT TO ACHIEVE (ONE OR MORE) SUSTAINABILITY OBJECTIVES

The tool optimises a given building design by varying its design variables, until minimum (embodied, operating or both) energy consumption for given flexibility of building is achieved.

Use Case

USE-4 As shown in (Figure 4.5), UC-1 (or UC-3) uses the tool to optimise a building design to achieve minimum energy consumption during its life-cycle. Alternatively, UC-2 (or UC-3) uses the tool optimise a building product to achieve minimum energy consumption during its life-cycle.

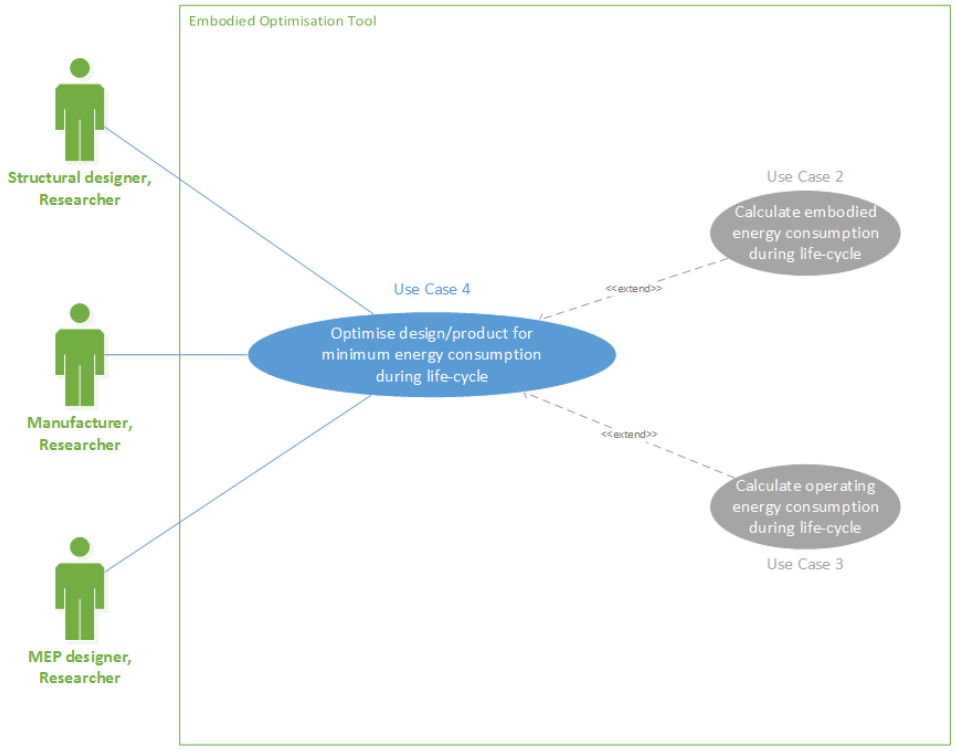

Figure 4.5: Use Case 4 - Optimise design or product to achieve (one or more) sustainability objectives

Functional Requirements

REQ-23 The user is provided with a suitable optimisation algorithm to to minimise embodied energy consumption or operating energy consumption of a building or maximise the adaptability of a building. This will in turn refer to their respective functional requirements REQ-09 to REQ-23. A combination of these three objectives should also be possible with the algorithm.

REQ-24 Allow the user to determine which design variables are to be varied during optimisation as well as their extent (upper and lower limits) of variation.

REQ-25 Provide user with at least one form of visualisation to study the final energy consumption results obtained.

EXPORT DESIGN TO OPERATE WITH THIRD-PARTY TOOLS

Aside from the above mentioned functions, a *building design* is also used by other software tools for project management, detailed structural calculations, quantitative analysis, production, etc. Therefore, it is desired that the tool exports a given building design to interoperate with at least one third-party tool.

Use Case

USE-5 UC-1 (or UC-3) uses the tool to export a building design to a contractor for production purposes, or to a structural engineer for detailed structural calculations of high-loaded areas or connections, etc (Figure 4.6).

Functional Requirements

REQ-26 The tool should be able to explode a building design to obtain a centerline model, together with identifying names and cross-section profiles.

REQ-27 Export the exploded design model to an Industry Foundation Classes (IFC) model as *.ifc files can be read or imported by several software in the building industry.

REQ-28 Export the exploded design model to work with at least one alternate structural analysis and design software.

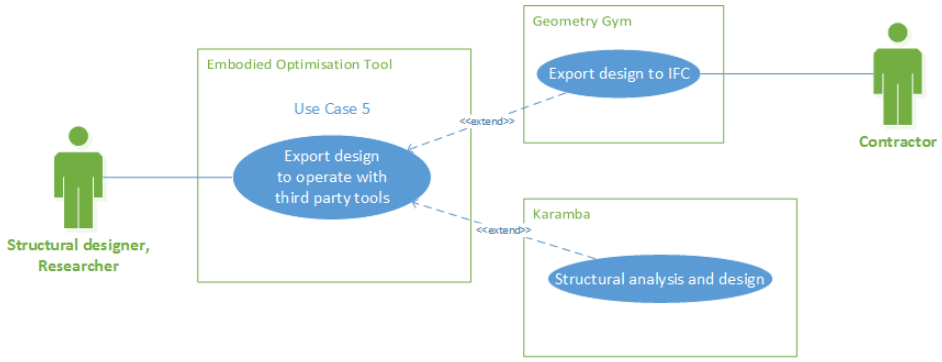

Figure 4.6: Use Case 5 - Export design to operate with third-party tools

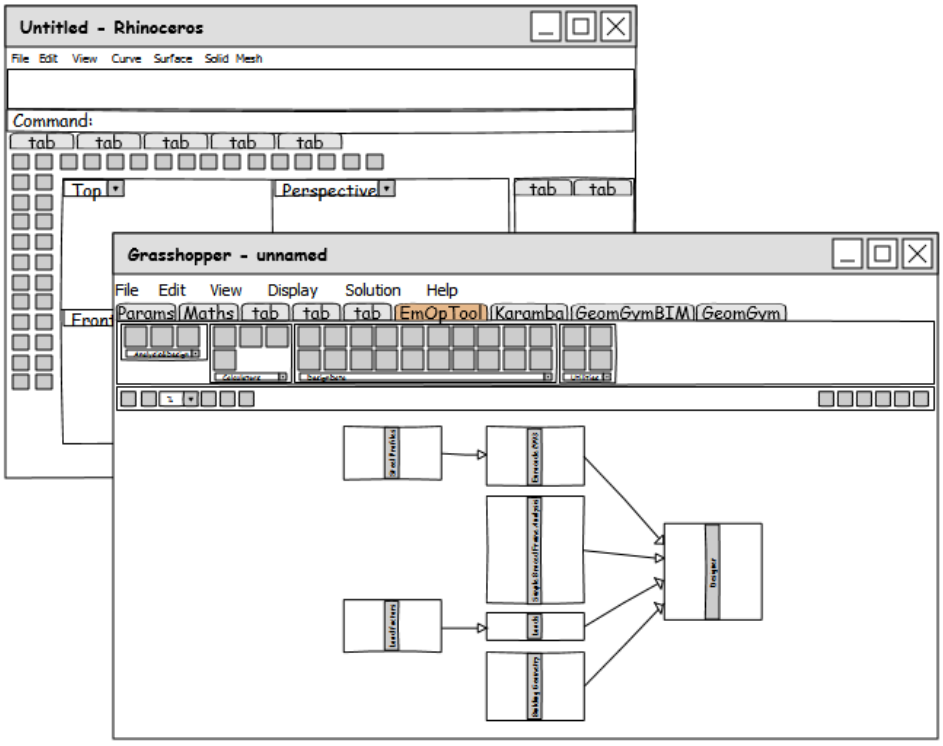

Figure 4.7: Mock up of user interface

4.2.4. EXTERNAL INTERFACE REQUIREMENTS

User interface The embodied optimisation tool is in the form of a Grasshopper plugin. It will be available as a tab within the Grasshopper user interface. The tab will contain the different components of the tool. The description of the different components have been provided in Appendix C and a mock-up of the interface is displayed in Figure 4.7. Suitable icons are to be provided for each component.

4.2.5. OTHER NON-FUNCTIONAL REQUIREMENTS

Software Quality Attributes The additional qualities desired in the tool are:

- *Adaptability*, so that the tool can be adapted for any kind of structure, building material, building code, analysis method, etc.
- *Availability*, such that the tool is made available as an open source sustainability platform, based on which others can contribute to further development of the tool.
- *Reliability* through visualisations and reports, the operations and calculations made are available to the user, with which the results obtained can be verified by the user.

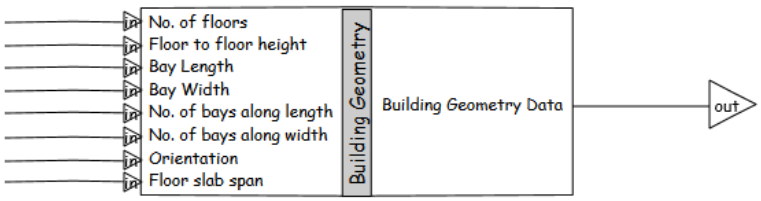

Figure 4.8: Mock up of Grasshopper component "Building Geometry"

4.3. DESIGN

Design of a software evolves as a project progresses. A preliminary design was made based on the above requirement specifications. Based on findings/observations during the development process as well as advice from the project committee, this design was improved several times over the course of the project. Below is a description of the final design adopted by the tool.

4.3.1. STANDARD LANGUAGES & TOOL

The embodied optimisation tool is developed in *C# (C-sharp)* programming language. Using *Microsoft Visual Studio* integrated development platform (IDE), the tool is integrated with the 3-D modeling tool, Rhinoceros and its plug-in Grasshopper.

C# (C-SHARP)

A programming paradigm is a style of computer programming. C# is a multi-paradigm programming language, thus providing a powerful framework for the programmers in which they can work in a variety of programming styles by intermixing different paradigms. C# is developed by Microsoft and is designed for the Common Language Infrastructure. Mono (open-source), DotGNU (open-source) and .NET Framework (Microsoft) are the three major implementations for C# (Drayton et al. 2003). The main benefits of programming in C# is as follows:

- Multi-paradigm (one of them being object-oriented);
- Intended to be simple, modern and general-purpose;
- Aims to provides support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection;
- Intended for use in developing software components suitable for deployment in distributed environments;
- Gives importance to support for internationalization, software robustness, durability, programmer productivity, source code portability and programmer portability;
- Suitable for writing (very large to very small) applications for both hosted and embedded systems;
- Intended to be economical with regard to memory and processing power requirements (though not at the same level as C or assembly languages) .

4.3.2. ACHIEVING THE FUNCTIONAL REQUIREMENTS

The functional requirements of the tool were listed earlier in the chapter. Before beginning the development, it is necessary to determine how each of these requirements can be achieved.

DESIGN A BUILDING

REQ-01 The required design data is classified into suitable categories. Each category is represented as a (Grasshopper) *component*, which accepts the relevant data from the user. This data is wrapped together and passed as an output parameter from the component, for further operation/calculation. For example, Figure 4.8 illustrates the data categorised as *Building Geometry*. Class diagrams are drawn for all the design data categories, this can be viewed in the Appendix C.

Type	Programming	Interface
Text (.txt)	Simple	Not suitable for organising and storing large amount of data
Comma separated value (.csv)	Simple	Between text and spreadsheet files
Spreadsheet (.xlsx or .ods)	Complex	User friendly
Internal database	Complex	Closed off from end-user

Table 4.1: Comparison of different database options

REQ-02 Design data are input either in the form of manually enteries or as a multiple-choice menu. In Grasshopper, data can be entered manually by the user via primitive Grasshopper components such as *Sliders, Integer parameter, Number parameter* or *Text panels*. Multiple choice inputs are appended to the component's menu (obtained by right-clicking the particular component), and the user can *check* the desired option.

REQ-03 A comparison was made of different options (Table 4.1) and comma separated value file format was selected. Steel profile data for European profiles: IPE (for beams), HE(A,B,M - for columns) and CHS (for wind bracing) are saved as a .csv file. The first row of the .csv file contains *column names* such as *shape, name, mass, outer diameter, height,* etc. which acts as identifiers for the tool to access the right data for calculations. By following this format, the user can easily add new profiles information (say, the British UB or UC) to the .csv file and the tool is able to read and use this data correctly. Following the computational framework for sustainable building design, an interface named `IMaterialProfiles` is created for the tool. This interface contains methods to correctly add or provide the steel data to other components of the tool. This interface is to be implemented by all material-profile components in the tool. In this case, this is the *Steel Profiles* component, which accepts the file path of the .csv file from the user.

REQ-04 Guided by the computational framework, interfaces `IDesignCode` and `IAnalysis` are created for the tool, which are to be implemented by any design code component and analysis method component respectively. As determined in the project objective, these are the *Eurocode 1993* component and *Simple Braced Frame Analysis* component. `IDesignCode` contains methods which output material profiles for structural elements that satisfy the ultimate limit state and serviceability limit state design checks of Eurocode. The respective design forces acting on the particular structural element (primary or secondary beam or column or wind bracing) are to be calculated and provided by the methods of IAnalysis. Both the interfaces also contain methods to provide the calculation steps to the user in the form of *Analysis Report* and *Design Report*. Together, these two interfaces lay the skeletal framework for easily adding or modifying design codes and analysis methods in the tool.

REQ-05 A *Designer* component accepts the building geometry, loads, design code and analysis methods as inputs and generates a building design and calculations report as outputs. The building design is to be in the format of `IMaterialProfiles`, which can be accessed by other components. The component by itself does not perform the design, instead it calls on the various methods of IDesignCode and IAnalysis. In this manner, a change in the design code or analysis method does not affect the functioning of this component, provided that the respective interfaces are implemented correctly.

REQ-06 The calculation report is provided by the *Designer* component. However, this report is text within the Grasshopper environment, which can be can be copied and saved elsewhere by the user. Alternatively, a *Report Generator* component is to be created which calls methods from `itextsharp.dll` library to export this text into a portable document format (.pdf) file.

REQ-07 An *Explode Model* component is to be created, which accepts the building design (in the form of `IMaterialProfiles`) and building geometry to generates a 3D centerline model and the actual names of the designed section profiles, together with identifier names to relate to their position or location in the 3D model.

REQ-08 A Finite Element Analysis plug-in for Grasshopper is Karamba3D (2014). A component *Model From Karamba* is to be created, to convert an analysed building design from Karamba into `IMaterialProfiles`.

Material	Embodied Energy	Embodied Carbon	Source
Structural Steel	7.3 MJ/kg	0.48 kg-CO2/kg	Bouwen Met Staal, NL
Steel Plate	11.7 MJ/kg	0.791 kg-CO2/kg	SAB Profiel Sandwich Panelen, NL
PIR Insulation	72.1 MJ/kg	3 kg-CO2/kg	SAB Profiel Sandwich Panelen, NL
Stonewool Insulation	16.8 MJ/kg	1.05 kg-CO2/kg	SAB Profiel Sandwich Panelen, NL
Fibre Glass	28 MJ/kg	1.53 kg-CO2/kg	ICE by University of Bath, UK
Concrete (1:1.5:3)	1.11 MJ/kg	0.159 kg-CO2/kg	ICE by University of Bath, UK
Plywood	15 MJ/kg	0.81 kg-CO2/kg	ICE by University of Bath, UK
Photovoltaic (Monocrystalline)	4750 MJ/m2	242 kg-CO2/m2	ICE by University of Bath, UK
Photovoltaic (Polycrystalline	4070 MJ/m2	208 kg-CO2/m2	ICE by University of Bath, UK
Photovoltaic (Thin film)	1305 MJ/m2	67 kg-CO2/m2	ICE by University of Bath, UK

Table 4.2: Embodied Energy Values used by the tool

CALCULATE EMBODIED ENERGY CONSUMPTION

REQ-09 Similar to steps followed in achieving REQ-01.

REQ-10 Embodied energy data of different building materials are collected for Western Europe, particularly Netherlands (Table 4.2) and saved as a .csv file. The *Embodied Energy Calculator* component should accept the file path of this file, from the user.

REQ-11 Similar to steps followed in achieving REQ-02.

REQ-12 Designing a building is covered in REQ-01 to REQ-08. The designed material profiles output by the *Designer* component serves as input into the *Embodied Energy Calculator* component, which obtains the number of profiles and their weights via methods of `IMaterialProfiles`.

REQ-13 The *Floor Systems* component accepts from the user, the type of floor system and the percentage of base material (steel, concrete, timber) present in the floor system and the building geometry. With this information, total weight of material can be determined by the component and provided to the *Embodied Energy Calculator*. Similarly, a *Façade* component accepts façade related information from the user, which is then accessed by the the *Embodied Energy Calculator* component, to determine the type of base materials (PIR, stonewool, steel plate or glass) and their quantities.

REQ-14 Embodied energy consumption of the different parts of a building is calculated by the *Embodied Energy Calculator* component based on the location of the building, type of materials and their quantities used. This is the initial embodied energy consumption. The skin of the building has a life of 15-30 years and is expected to undergo changes during the life of the building. The number of years after which the change occurs is input by the user. Based on this value, the energy calculator adds additional energy consumption for every change in façade or roof material, thereby resulting in the total embodied energy consumption of the building over its life-cycle. 10% extra energy is also added to compensate for Grey and Induced energy consumption. The component should also output the separate energy values in combination with the time-line when they are introduced in the building. This output is handy to plot and study the total life-cycle energy consumption of the building.

REQ-15 Similar to steps followed in achieving REQ-01.

REQ-16 The *Adaptability Rater* component accepts inputs on the various adaptability aspects. Based on these inputs, estimated service life (ESL) factors are calculated for each aspect. Multiplying the ESL factors with the life of the building provides the estimated service life, which can then be input into the *Embodied Energy Calculator* component. The individual factors are interpolated from the graphs and tables provided in Appendix A.

REQ-17 The user can also replace the estimated service life calculated by the *Adaptability Rater* component with a suitable number input.

CALCULATE OPERATING ENERGY CONSUMPTION

REQ-18 Similar to steps followed in achieving REQ-01.

REQ-19 Similar to steps followed in achieving REQ-02.

REQ-20 Annual weather data for the building location is saved as a .csv file. The user provides the *Weather Data* component with the file path for this file. The first row of the .csv file contains *column names* such as *maximum temperature, minimum temperature, daytime humidity,* etc. which acts as identifiers for the tool to access the right data for calculations. By following this format, the user can easily add or replace old weather data in the .csv file and the tool is able to read and use this data correctly.

REQ-21 The operating energy calculation method is for low rise office buildings, based on the Building Energy Tool developed by Arup (New York). This method is implemented into *Operating Energy Calculator* component. The component calculates separately the yearly energy consumed by heating, cooling, fans, pumps, hot water, lighting, equipments, vertical transport, humidification and the energy generated by photovoltaic in the building, if any. From these values, it obtains the the annual operating energy consumption of the building.

REQ-22 To determine the operating energy consumption over the life cycle, the annual energy consumption can be multiplied to the estimated service life using Grasshopper's multiplication component.

OPTIMISE DESIGN OR PRODUCT TO ACHIEVE (ONE OR MORE) SUSTAINABILITY OBJECTIVES

REQ-23 Grasshopper comes with in-house optimisation component *Galapagos* (Rutten 2014), which provides the user with the option to choose from two different optimisation algorithms - the *Genetic solver* and the *Simulated Annealing solver*. Another option for optimisation is via the *Octopus* plug-in (Vierling 2014).

REQ-24 Both Galapagos and Octopus performs optimisation by varying the values in *Sliders* connected to them. The user can set the upper and lower limits for these *Sliders*, such that only the values that fall within the range are used during optimisation.

REQ-25 A *Cumulative Energy Plotter* component is to be created to plot a 2D graph of the total cumulative energy consumption of a building during its life-cycle. The energy values output by *Embodied Energy Calculator* and *Operating Energy Calculator* are input into this component, together with the estimated service life and suitable scaling factors for plotting the graph.

EXPORT DESIGN TO OPERATE WITH THIRD-PARTY TOOLS

REQ-26 Covered in REQ-07.

REQ-27 Geometry Gym's IFC plug-in for Grasshopper support the creation of IFC models (Mirtschin 2014). Trial license can be obtained by contacting the developer. The *Explode Model* component provides inputs for Geometry Gym IFC components.

REQ-28 Covered in REQ-08, with Karamba plug-in for Grasshopper.

4.3.3. SEQUENCE DIAGRAMS

It is useful to draw sequence diagrams to indicate the work-flow within the tool, in the sequential order that the various interactions occur. As mentioned in subsection 4.2.5, adaptability is one of the desired non-functional attribute in the tool. This implies that the tool can be adapted for any kind of structure, building material, code, analysis method, etc. This is also one of the main feature of the parametric framework for sustainability, as indicated in the beginning of this chapter (Figure).

Accordingly, a sequence diagram was developed for the functional requirement *Design a building*, such that each step of the design is handled by a separate module, which can be easily replaced with suitable alternatives. Figure 4.9 indicates how the *Designer* component links the various design data with the analysis method to obtain the design forces acting on the structure, which is then linked to the building code to obtain suitable material profiles that form the design of the building.

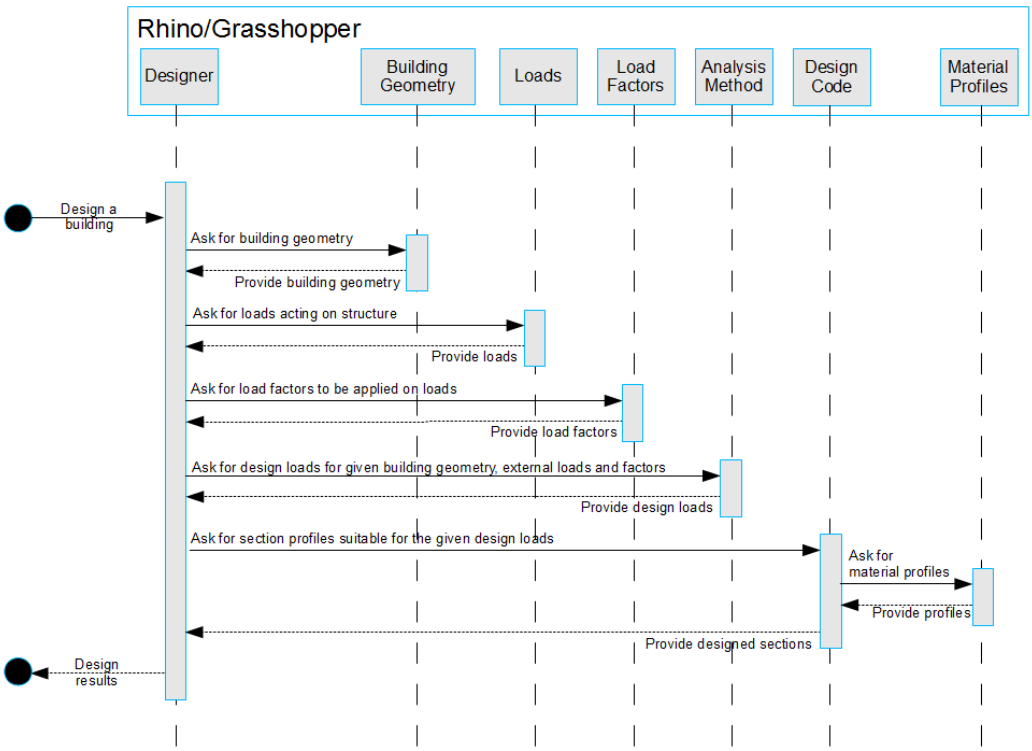

Figure 4.9: Sequence diagram illustrating the interactions during design of a building

4.3.4. CLASS DIAGRAMS

Class diagrams are also created for each component of *Embodied Optimisation Tool* to illustrate the attributes and operations for each class involved in making of the components, including any relationships between different classes. A class diagram template is provided in Figure 4.10, which is to be implemented by most classes in the tool. The complete list of class diagrams can be viewed in Appendix C, together with a list of all available components in the tool.

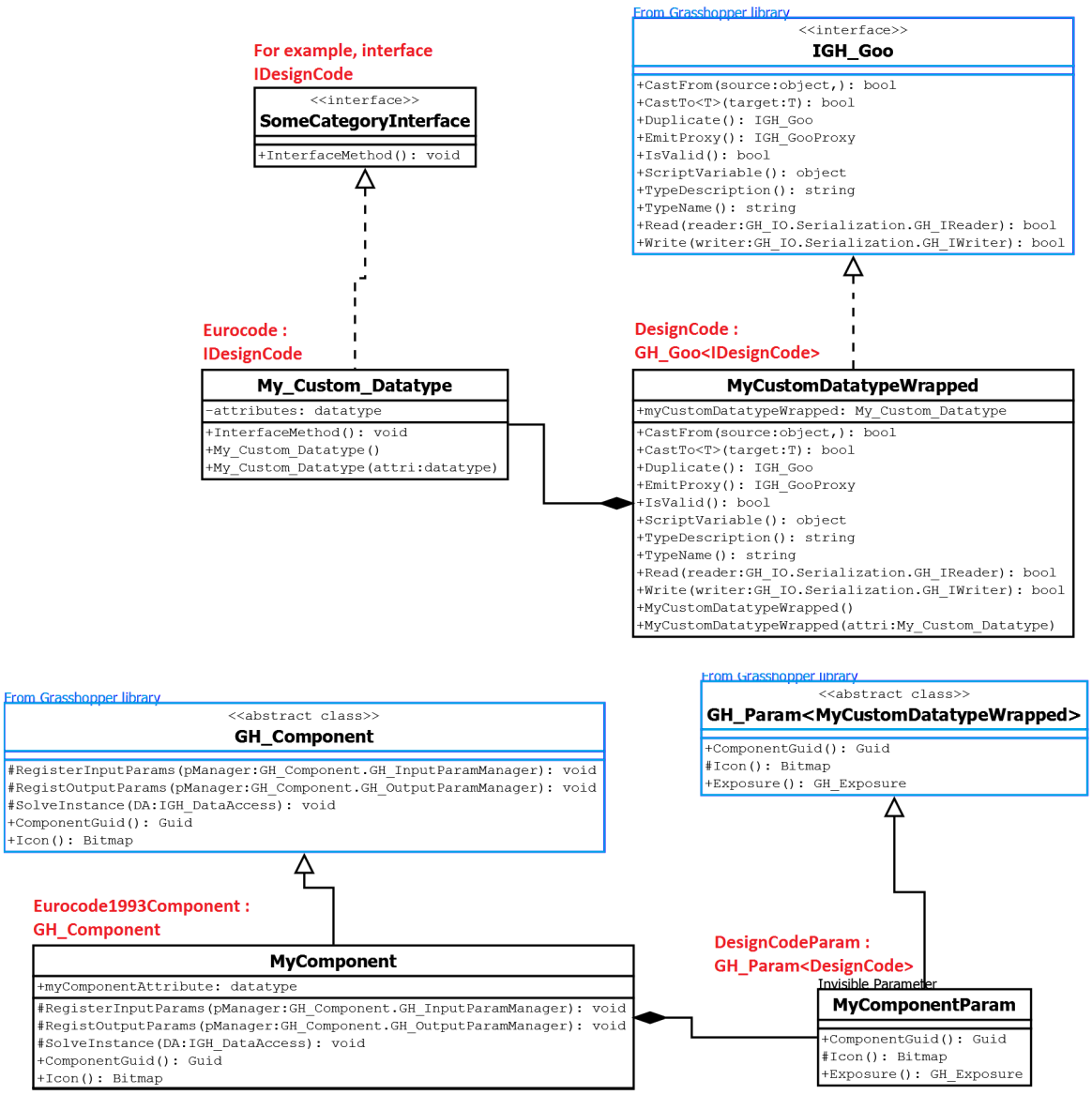

Figure 4.10: Class diagram template implemented by most classes in the tool

5

Development of Embodied Optimisation Tool

The various requirements of *Embodied Optimisation Tool* were determined in the previous chapter. This chapter follows the implementation of these requirements during the course of development of the tool. The chapter explains the development methodology followed in this project and the various test cases that were developed to validate the tool, and the discuss the results obtained.

5.1. Development Methodology

Several methodologies (and their combinations) are available in the field of software development. The development phase of this tool is based on the test-driven methodology (Figure 5.1). In other words, the tool is divided into different features; which are then implemented one by one by first writing the test(s) for the feature, and then testing the tool for this feature. On failure, the feature is implemented and then tested again until it passes all the tests. Then the test for the next feature is created and the process continues until all features have been successfully implemented.

Based on IEEE's standards for testing, four kinds of tests are to be performed during *professional* software development:

- **Unit testing** verifies functioning of each module of the feature;
- **Integration testing** verifies integration of all the modules within the feature;
- **New feature testing** verifies the integration of all the features;
- **Regression testing** verifies functioning of the old features, after addition of new feature.

However this is time-consuming and/or requires large man-power, and therefore, is not strictly enforced in the development of *Embodied Optimisation Tool*. Instead, several test cases were developed in order to test the functionalities of the tool after the program is run (dynamic testing). These test cases consist of a combination of integration testing, new feature testing and regression testing to a certain extent. Some unit testing was also carried out during the programming stage of the tool. The test cases used have been presented in the last section of this chapter.

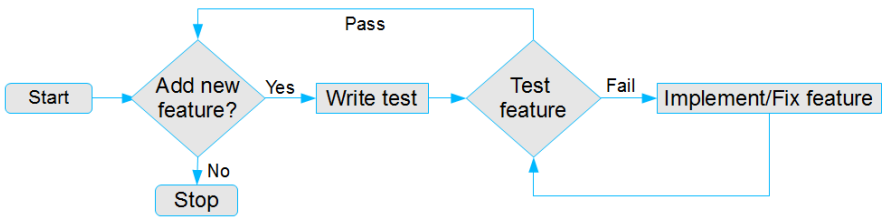

Figure 5.1: Test-driven development methodology flowchart

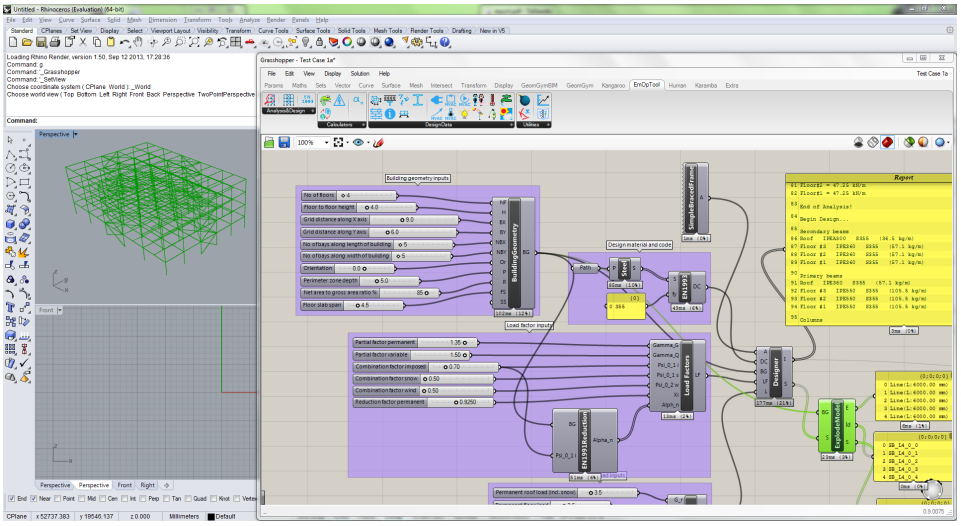

Figure 5.2: GUI of *Embodied Optimisation Tool*

5.2. GRAPHICAL USER INTERFACE (GUI) OF TOOL

The general appearance of the tool is based on the GUI provisions from Grasshopper, as indicated in the GUI mock up (Figure 4.7). Within this GUI, icons have been designed for each of the component. Also, the various components are categorized based on their role in the tool, into:

- Analysis and design components;
- Calculators;
- Design data components;
- Utilities.

Figure 5.2 shows the final GUI of the tool prototype after development.

5.3. TESTING & VALIDATION OF TOOL

The *Embodied Optimisation Tool* has 5 main features, as determined and described in the previous chapter. Each of these features require testing in order to render the tool as functional and validation of these test results to ensure that the tool can be trusted. Test cases were prepared for each feature, which are described in the following paragraphs, together with the observations made and results obtained. The test cases are in the form of an ensemble of components connected to each other, and is called a *Grasshopper script*.

5.3.1. DESIGN A BUILDING

This test case aims at testing REQ-01 to REQ-08.

DESIGN BUILDING USING EMBODIED OPTIMISATION TOOL COMPONENTS

The first seven requirements aim at designing the building using components of Embodied Optimisation Tool. Figure 5.3 shows a portion of the Grasshopper script used.

Description and Observations In this script, the design data is input into the *Building Geometry* component, *Loads* component and *Load Factors* component. File path of .csv file containing European steel profiles is provided to the *Steel Profiles* component. This data is then passed into the *Eurocode 1993* component, which also accepts the steel grades for these material profiles. To calculate the forces, the *Simple Braced Frame Analysis* component is used. The output from each of the above mentioned components is connected as inputs into the *Designer* component, which in turn outputs the calculation report and the designed section profiles. With the aid of Grasshopper's provisions for warning and error messages, care is taken to ensure that the user does not forget any inputs or enter wrong inputs (in certain cases). The calculation report is

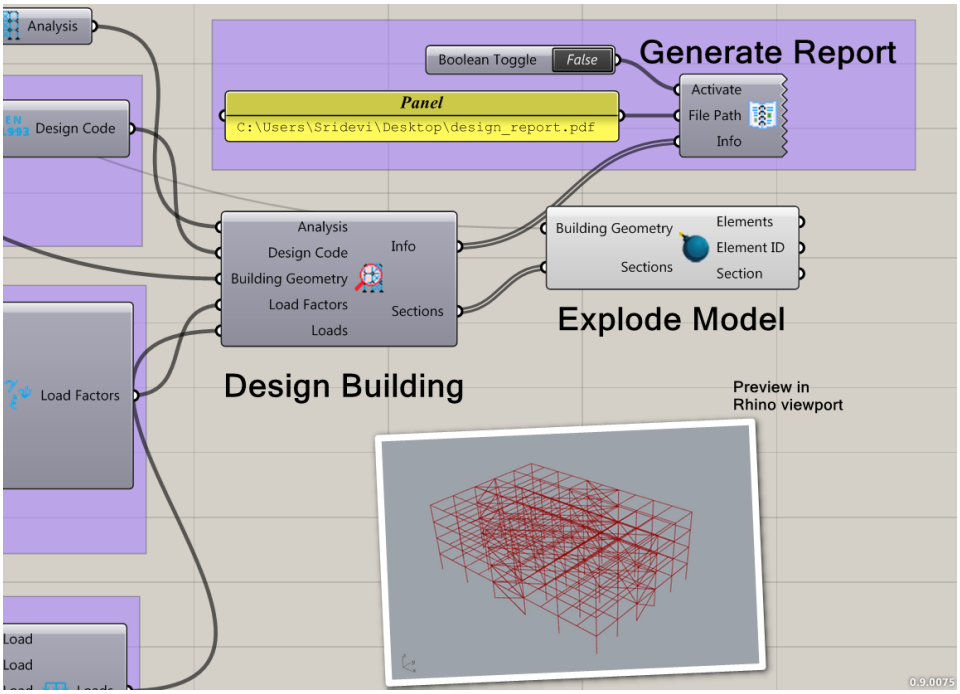

Figure 5.3: Test case for designing a building using Embodied Optimisation Tool

exported to pdf using the *Report Generator* component (Figure 5.4). The design is then exploded using the *Explode Model* component, which provides a 3D centerline model of the building design in the Rhino viewport, as well as identifiers and positions for each structural element designed by the tool. The end results per requirement have been tabulated in Table 5.1.

Requirements	Status	Remarks
REQ-01	Pass	-
REQ-02	Pass	-
REQ-03	Pass	Steel material properties hard-coded in component, profile data in .csv file
REQ-04	Pass	-
REQ-05	Pass	-
REQ-06	Pass	-
REQ-07	Pass	Tested successfully for rectangular grid building only

Table 5.1: Test case to design using Embodied Optimisation Tool components

Validation of feature The tool currently employs simple braced frame analysis method for calculating the forces for beams and columns, based on the example from The Steel Construction Institute (Appendix D). Design checks for the tool are based on the Dutch Annex of the Steel Eurocode. Due to the small impact of wind bracing on the total energy consumption of the building, elaborate design calculations were omitted for wind-bracing diagonals, instead replaced with simple geometrical calculations for forces and only tensile force check from the Steel Eurocode. In order to check the accuracy of calculations performed by the tool, the design data from the aforementioned example were input into the tool components, together with a csv.file containing British steel profile sections. A comparison of the results (Figure 5.5) prove to be very similar (complete calculations in Appendix D). Alternatively, the same building can be designed with any third-party structural design software and the results compared. This is performed as a part of test case for REQ-08 (below).

Design load for secondary middle beams supporting:
Roof = 20.25 kN/m
Floor#3 = 47.25 kN/m
Floor#2 = 47.25 kN/m
Floor#1 = 47.25 kN/m

End of Analysis!

Begin Design...

Secondary beams
Roof IPEA300 S355 (36.5 kg/m)
Floor #3 IPE360 S355 (57.1 kg/m)
Floor #2 IPE360 S355 (57.1 kg/m)
Floor #1 IPE360 S355 (57.1 kg/m)

Primary beams
Roof IPE360 S355 (57.1 kg/m)
Floor #3 IPE550 S355 (105.5 kg/m)
Floor #2 IPE550 S355 (105.5 kg/m)
Floor #1 IPE550 S355 (105.5 kg/m)

Columns
Roof HEAA160 S355 (23.8 kg/m)
Floor #3 HEAA240 S355 (47.4 kg/m)
Floor #2 HEAA280 S355 (61.2 kg/m)
Floor #1 HEAA320 S355 (74.2 kg/m)

Wind bracing
Diagonals along length of building CHSH101.6x3.2 S355 (7.8 kg/m)
Diagonals along width of building CHSH88.9x6.3 S355 (12.8 kg/m)

Calculation in Detail:

Design of Secondary beams
> ULS design Load = 20.42 kN/m
> Beam span = 6 m
> Design shear force = Design load * Beam span / 2 = 79.25 kN
> Design moment = Design load * Beam span ^ 2 / 8 = 118.88 kNm
> Yield strength = 355 N/mm2
> Partial factor = 1
> Plastic modulus required = design moment * partial factor / yield strength = 334.87 cm3
> Maximum allowed deflection = Beam span / 250 = 24 mm
> SLS design Load = 20.25 kN/m
> Choosing section profile IPEA300
> Cross-section class of section profile = 1
> Height of section profile = 297 mm
> Web-thickness of section profile = 6.1 mm
> Moment of inertia Y of section profile = 71730000 mm4
> Plastic modulus Y of section profile = 555.45 cm3 >= Plastic modulus required, therefore, OK
> Shear resistance of section profile = 1.04 * Height * Web-thickness * (Yield strength / 1.732) * Partial factor = 386.18 kN >= Design shear force, therefore, OK
> Deflection in beam = 5 / 384 * SLS design load * Beam span ^ 4 / Modulus of elasticity * Moment of inertia Y = 22.69 mm <= Maximum allowed deflection, therefore, OK

Figure 5.4: Sample page from design calculations report (pdf)

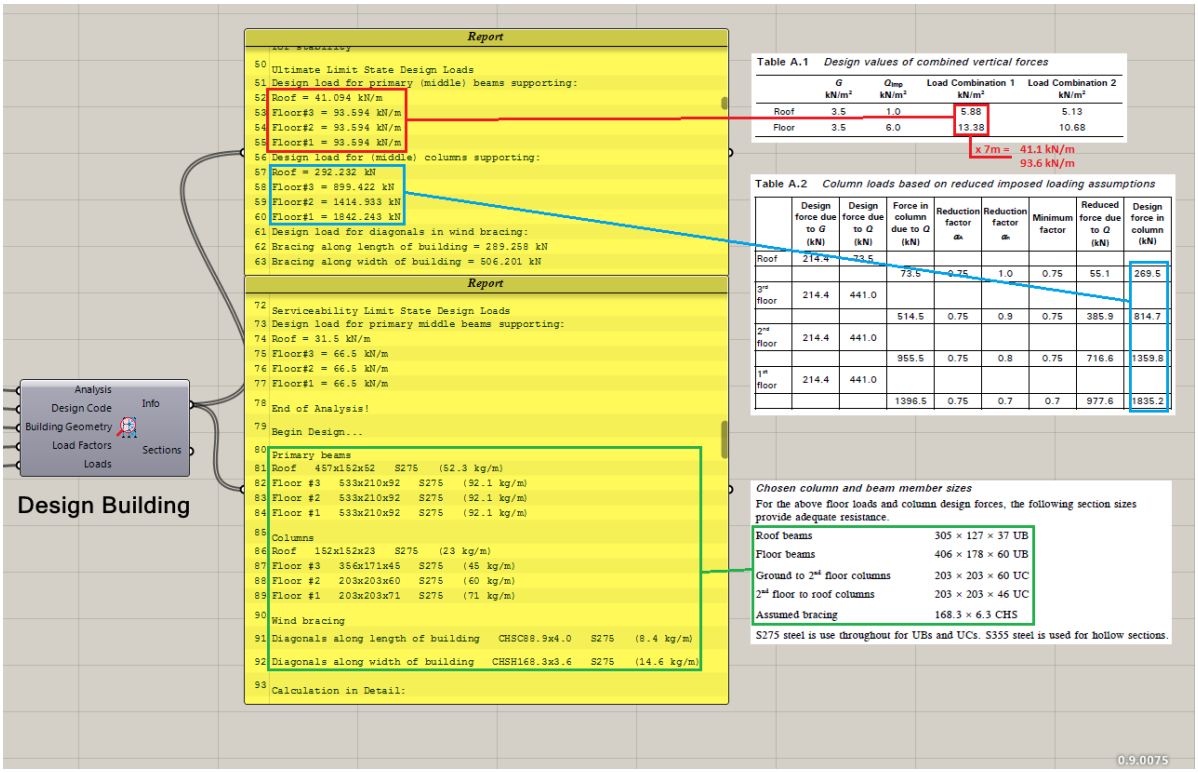

Figure 5.5: Comparison of design calculations between Embodied Optimisation Tool and example from Steel Construction Institute

IMPORT DESIGN FROM A THIRD-PARTY TOOL

REQ-08 aims at overriding the default design calculations with that of a third party tool and importing this design into *Embodied Optimisation Tool* for further action, as shown in Figure 5.6.

Description and Observations Karamba components are used to design the building using the same design inputs. The designed model is then input into *Model From Karamba* component, which converts it into *Embodied Optimisation Tool* format, identical to that obtained from the *Designer* component. The design is then exploded using the *Explode Model* component, which provides a 3D centerline model of the building design in the Rhino viewport, as well as identifiers and positions for each structural element designed by the tool.

Requirements	Status	Remarks
REQ-08	Pass	Tested successfully Karamba

Table 5.2: Test case to import design from third-party tools

Validation of feature The building was designed with both components from Embodied Optimisation Tool and with components from Karamba, for the same set of design inputs. Small differences are expected (Figure 5.7) since the force calculations in Embodied Optimisation Tool are based on loads acting on simply supported b eams at each floor level and a simple load take down for columns, whereas Karamba performs Finite Elements Analysis on the centerline model. It is also to be noted that refining the analysis mesh will provide better results in Karamba. However, the differences in values are comparable and therefore, the tool results can be deemed satisfactory.

5.3.2. CALCULATE LIFE-CYCLE EMBODIED ENERGY CONSUMPTION

This test case aims at testing REQ-09 to REQ-17. Figure 5.8 shows the key parts of the Grasshopper script used in order to calculate the life-cycle embodied energy consumption of a building. REQ-09 to REQ-14

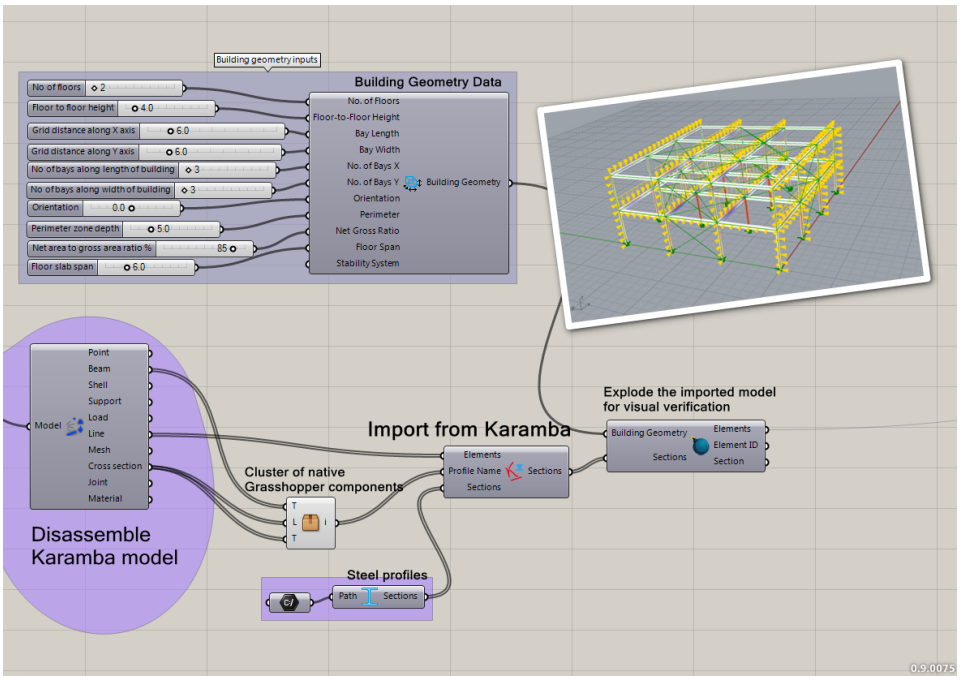

Figure 5.6: Test case for importing a design from Karamba

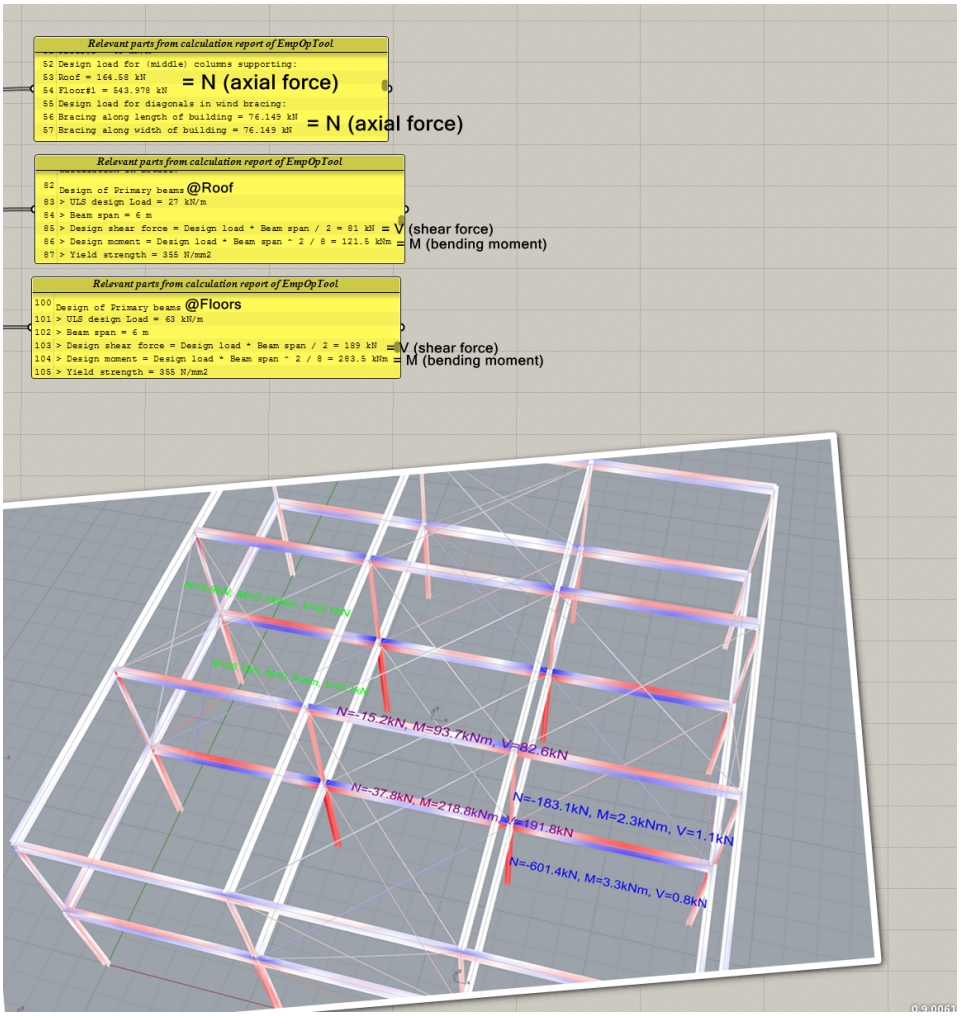

Figure 5.7: Comparison of design calculations between Embodied Optimisation Tool and Karamba

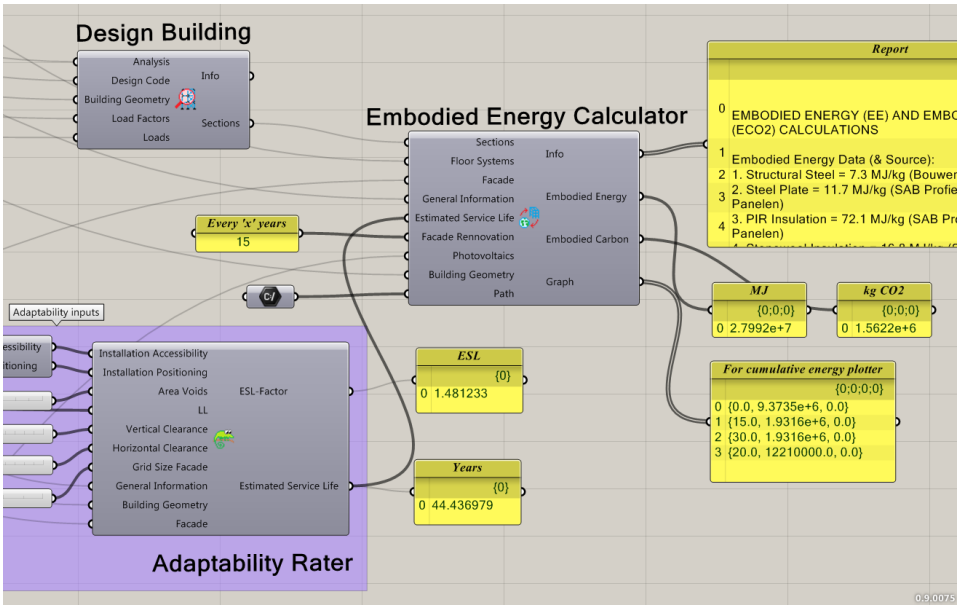

Figure 5.8: Test case to calculate life-cycle embodied energy consumption

covers calculation of embodied energy, while REQ-15 to REQ-17 aims at determining the adaptability rating of the building.

Description and Observations In this script, different sources of embodied energy consumption are input into the *Embodied Energy Calculator* component. This consists primarily of the material profile sections for load bearing structure as obtained from the output of the *Designer* component or *Import From Karamba* component. Other material inputs are the *Floor systems* component, *Façade component*, and *Photovoltaics* component. The *General Information* component collects inputs from the user regarding the location and function of the building and its (expected) service life, which is then output to the *Embodied Energy Calculator* component. The embodied energy component also accepts additional inputs for the estimated service life of the building and the number of years after which the façade is renovated in the building. The estimated service life is output by the *Adaptability Rater* component, based on the various adaptability or flexibility aspects input by the user. The year of installation is also an input of the *Photovoltaics* component, which adds to the determination of *life-cycle* embodied energy consumption of the building. The Embodied Energy Calculator component also outputs a calculation report, in addition to the total embodied energy consumption and embodied carbon consumption of the building during its lifetime. In order to study the embodied energy consumption during the life of the building, these values are also provided as co-ordinates for plotting a graph.

Validation of feature The calculations for embodied energy is directly related to the quantity of each material used and can therefore be easily validated by hand calculations. For this purpose, the *Embodied Energy Calculator* component also provides a calculation report, to compare the results obtained.

Arup conducted a study on a 6 storey office building for 5 different floor options and for two-different column grids (Perkins & Tandler 2005). A comparison of the results obtained from Arup and from the *Embodied Optimisation Tool* are plotted in Figure 5.9. In order to make the results comparable, façade and roof elements embodied energy values were excluded from the calculations. The steel beams were also excluded from calculation, since it is assumed in Arup study that the beams are part of the floor options, or absent when the floor slab rests directly on top of the columns. Also, two different embodied energy values for steel were provided into the tool, the former being cradle-to-gate embodied energy (as considered in Arup study) and the latter being cradle-to-cradle embodied energy (to account for recycled steel).

Since the exact quantities (per square meter) of steel and concrete in the differ floor slab options are not specified in the study report, it was assumed that 90% of the total floor weight is due to concrete and the remaining 10% of the weight is from steel. However, this assumption does not hold true for composite floors with standard or castellated beams, where the steel beam is heavier than this 10% of total weight, particularly in the case of longer spans. This accounts for the lower embodied energy consumption for all composite floor

Requirements	Status	Remarks
REQ-09	Pass	-
REQ-10	Pass	Not all building materials included, only those relevant to current tool prototype
REQ-11	Pass	-
REQ-12	Pass	-
REQ-13	Pass	Foundation not implemented, but can be approximated with *Floor System* component
REQ-14	Pass	
REQ-15	Pass	-
REQ-16	Pass	-
REQ-17	Pass	-

Table 5.3: Test case to calculate life-cycle embodied energy consumption

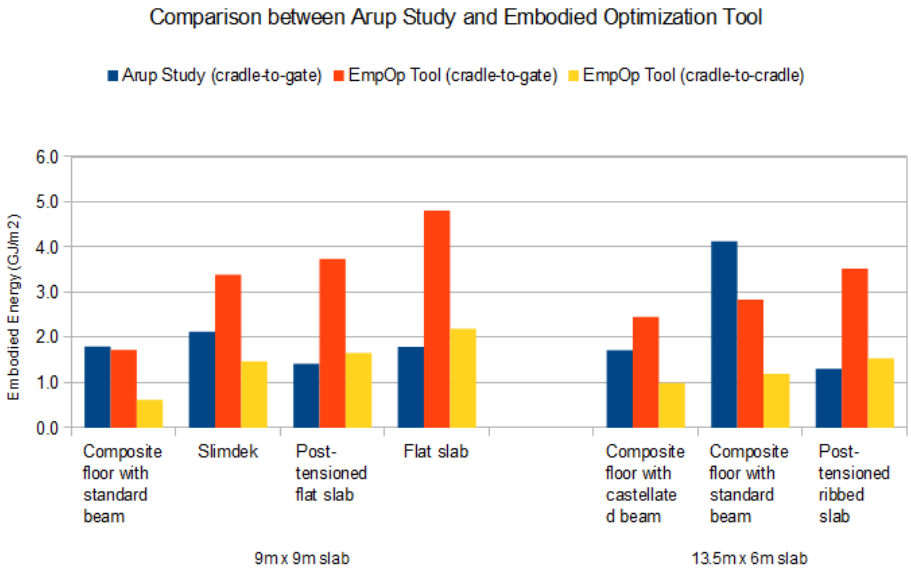

Figure 5.9: Comparison between structural embodied energy consumption for different floor options, as per Arup study and Embodied Optimisation Tool

solutions obtained from *Embodied Optimisation Tool*, when compared to their Arup study counterparts.

Also to be noted is that in the Arup study, the columns are made of concrete, whereas the *Embodied Optimisation Tool* designs steel columns. Therefore, in the Arup study, Slimdek and composite floor options have higher energy consumption when compared to the concrete floor options (which do not contain steel beams or deck in it). However, what is most interesting to note is that despite this big difference, when cradle-to-cradle embodied energy value for steel is used in the tool, there is a sizable drop in the total embodied energy consumption of the building, so that the higher steel content floor options - composite floor and Slimdek results in much lower energy consumption than their Arup study counterparts.

Perkins & Tandler (2005) concluded that it is necessary to study embodied energy over the complete life-cycle of the building, and that while the load bearing structure accounts for a significant part of the embodied energy, it is less significant than items like façade, finishes, etc., which are replaced during the life of the building. In this respect, the *Embodied Optimisation Tool* is able to provide much more insight into the total energy consumption of the building.

Adaptability Rater component is based on tool developed and validated by Frank Tool (2010). The *Embodied Optimisation Tool* implements its functionalities, in order to study the results of its combined optimisation, together with embodied energy calculations. Since the calculation of the ESL factor is based on interpolation of curves and simple mathematics, validation was not deemed necessary for this component.

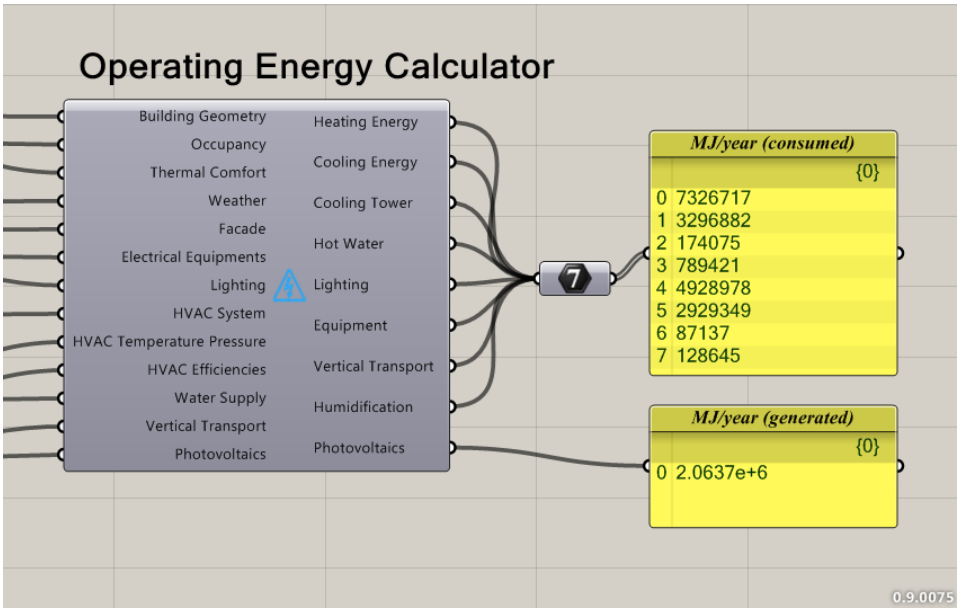

Figure 5.10: Test case to calculate life-cycle operating energy consumption

5.3.3. CALCULATE LIFE-CYCLE OPERATING ENERGY CONSUMPTION

This test case aims at testing REQ-18 to REQ-22. Figure 5.10 shows the key component in the Grasshopper script used in order to calculate the life-cycle embodied energy consumption of a building.

Description and Observations In this script, the operating energy design data are input into *Occupancy* component, *Thermal Comfort* component, *Weather Data* component, *Electrical Equipments* component, *Lighting* component, three *HVAC* components, *Water Supply System* component and *Vertical Transport System* component. The outputs from each of these components are input into the *Operating Energy Calculator* component. Aside from these components, the outputs from *Building Geometry* component, *Façade* component and *Photovoltaics* component are also input into the *Operating Energy Calculator* component. The component calculates the annual operating energy consumption separately for heating energy, cooling energy, cooling tower energy, hot water energy, lighting energy, equipment energy, vertical transport energy and humidification energy. Photovoltaic energy is the energy generated by the building. From these values, the total operating energy consumed by the building can be obtained.

Requirements	Status	Remarks
REQ-18	Pass	-
REQ-19	Pass	-
REQ-20	Pass	Weather data in .csv file is for New York
REQ-21	Pass	Energy consumption due to HVAC fans and pumps not included
REQ-22	Pass	-

Table 5.4: Test case to calculate life-cycle operating energy consumption

Validation of feature The operating energy calculations is based on the *Building Energy Tool* for low to medium rise office buildings developed and validated by Arup. Figure 5.11 compares the results from the *Building Energy Tool* and *Embodied Optimisation Tool*, for the same set of input values. Both have identical results, except for the slight lower energy values for heating, cooling and cooling towers. This is because the section Fans & Pumps, which is not implemented in the tool, affects these energy values.

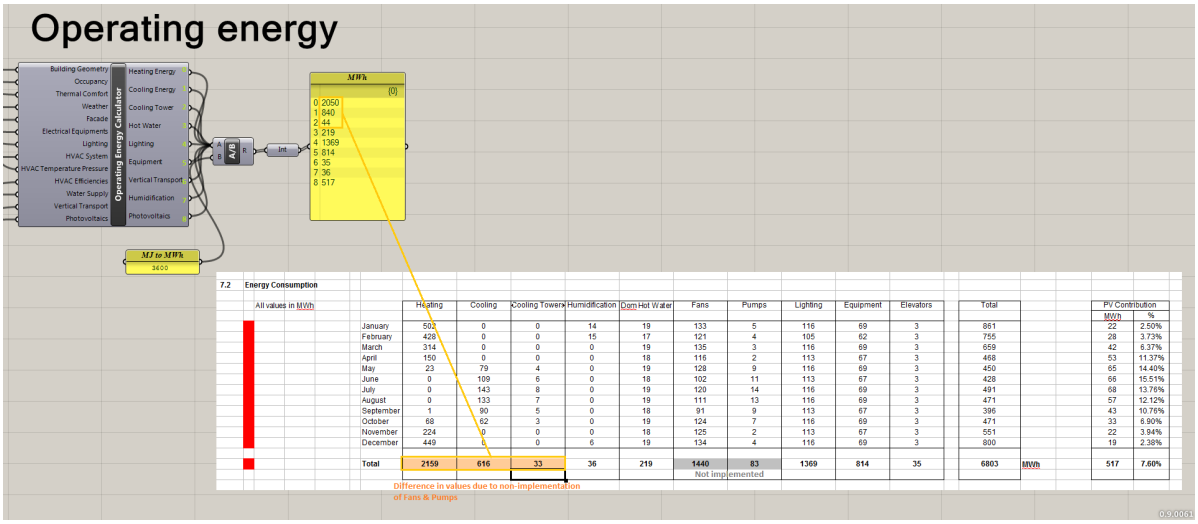

Figure 5.11: Comparison between results from Operating Energy Calculator component and Arup's Building Energy Tool

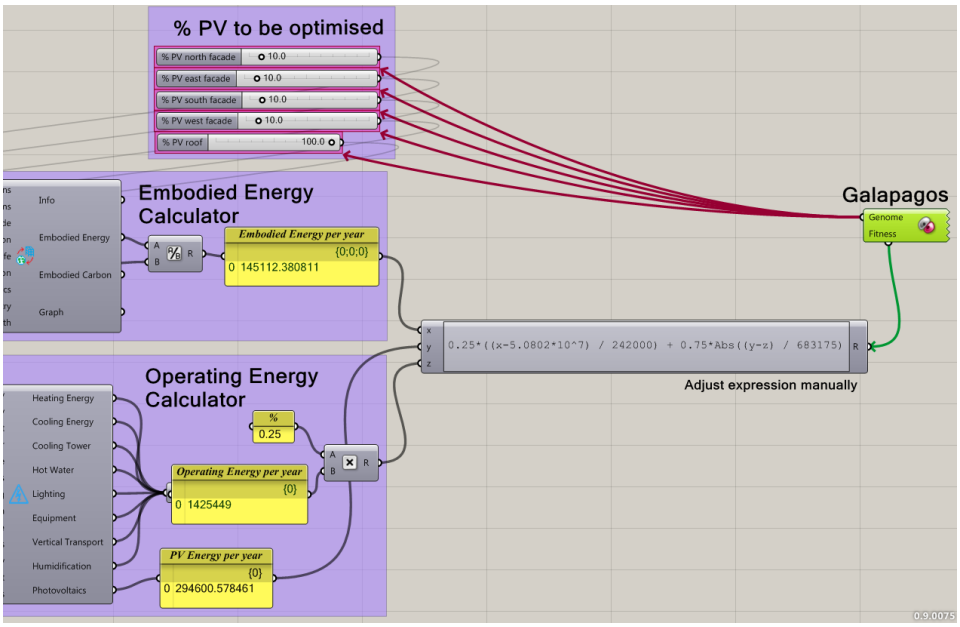

Figure 5.12: Test case to calculate life-cycle operating energy consumption

5.3.4. OPTIMISE DESIGN OR PRODUCT TO ACHIEVE (ONE OR MORE) SUSTAINABILITY OBJECTIVES

This test case aims at testing REQ-23 to REQ-25. Two different optimisation components available for Grasshopper: *Galapagos* (for single objective optimisation) and *Octopus* (multi-objective optimisation) were used with the tool. Hence, two test cases were developed, to demonstrate the capabilities of these two components in combination with the various components of *Embodied Optimisation Tool*. The Grasshopper scripts for both test cases are composed of the exact same set of components used in calculating embodied energy consumption and operating energy consumption. The only additional component is the optimisation component.

OPTIMISATION WITH GALAPAGOS

Figure 5.12 illustrates the application of *Galapagos* to obtain the percentage of photovoltaics to be applied to a building, in order to obtain a return of *x%* of annual energy consumption.

Description and Observations The genomes to be varied are connected to *Galapagos* component, which are in the form of *Slider* components only, with the boundaries of each slider providing the constraints for optimisation. The optimisation objective is a number parameter and is connected to *Galapagos* as well. Using

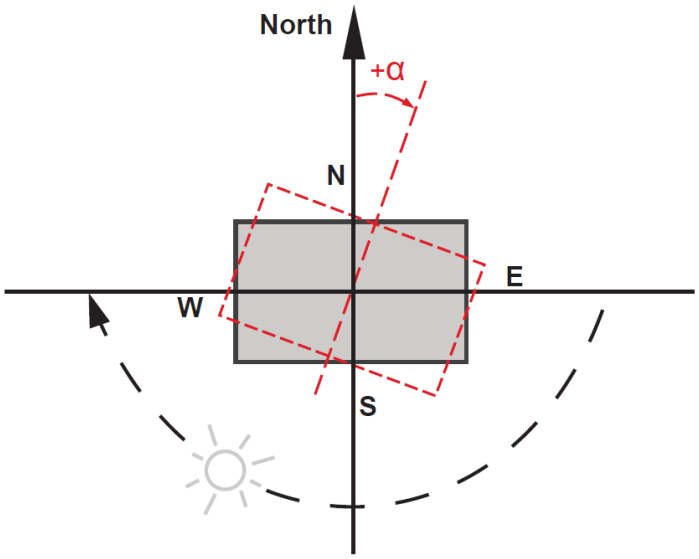

Figure 5.13: Orientation of Building with respect to true north

simple Grasshopper mathematics components, 25% of operating energy is determined and a comparison is made with the PV energy generated. When the difference between these two numbers are minimum (close to zero), the desired energy return is obtained from the photovoltaic. At the same time, the embodied energy should also be minimised to ensure that *Galapagos* provides only the minimum required amount of PV to achieve the target. To handle multiple objectives in *Galapagos*, it is recommended by the developer to create function to combine the different objectives (Rutten 2011). In the *Galapagos* window, target option 'minimise' is selected. The simulation is run until convergence. At this point, the sliders should be at the optimal position to obtain the desired 25% energy returns.

Validation of feature In order to ensure that the results obtained after optimisation with *Galapagos* can be trusted, the optimisation procedure is applied to a square building with three different orientation with respect to the sun, keeping the rest of the design inputs constant. The operating energy calculation tool measures the orientation of the building from the true north to the building north, with clockwise rotation resulting in positive orientation angles. In the northern hemisphere, the path of the sun is as indicated in Figure 5.13. Therefore, it is expected that a positive orientation angle will result in most sunshine on the roof, south and east façade, whereas a negative orientation angle will result in most sunshine on the roof, south and west façade. Therefore, the optimisation should also result in most amount of PV in these façades. The results obtained from the different optimisation run displays this behaviour, as shown in Table 5.5. However, *Galapagos* only provides a single optimum solution, which might not be the case always. This was observed at 0 degree orientation, where different optimisation runs provided different results for east and west façade(see Table 5.5), although the total amount of PV was the same. The user should therefore apply caution while using *Galapagos*, and validate the results obtained with outside logic.

OPTIMISATION WITH OCTOPUS

The U-value of insulation is inversely related to its thickness. Accordingly, it is useful to obtain the relationship between embodied energy (related to thickness of insulation) and operating energy (related to U-value of insulation) consumption of a building, for different U-values (and thicknesses). Also of interest is to understand whether a particular combination of U-values for façade and roof insulations will results in the most optimum solutions. *Octopus* is used to obtain the Pareto Optimal Front between embodied energy and operating energy for varying U-value (and thickness) of façade. Figure 5.14 illustrates the arrangement of the key components in the test script with *Octopus*.

Orientation	PV North	PV East	PV South	PV West	PV Roof	Desired Energy	Calc. Energy
-30 deg	0%	79%	100%	100%	100%	643095 MJ	642907 MJ
0 deg	0%	87.5%	100%	87.5%	100%	642574 MJ	642577 MJ
0 deg	0%	78%	100%	97%	100%	642574 MJ	642577 MJ
+30 deg	0%	100%	100%	79%	100%	643095 MJ	642907 MJ

Table 5.5: Optimisation Results from Galapagos, for different orientation of building at 10% generating efficiency

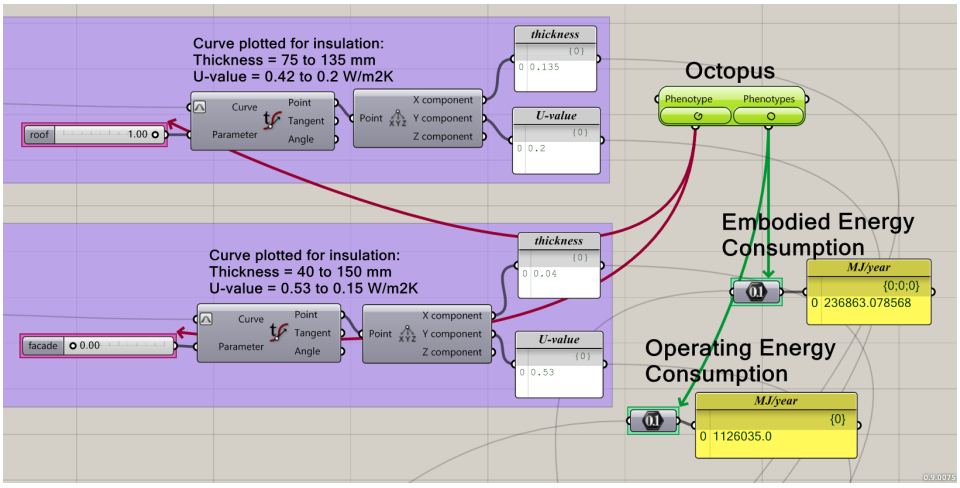

Figure 5.14: Test case to calculate life-cycle operating energy consumption

Description and Observations The description of this test case is similar to the test case with *Galapagos*. In addition, the graph in the *Octopus* window displays one of the generations during optimisation. Based on the results obtained, in combination with other factors such as cost, time, etc., the user is able to make a well-informed decision regarding insulation for the building.

Requirements	Status	Remarks
REQ-23	Pass	*Galapagos* output only single optimum solution, *Octopus* outputs Pareto Optimal Front
REQ-24	Pass	-
REQ-25	Pass	*Cumulative energy plotter* component takes 4-5 seconds to generate graph

Table 5.6: Test case to optimise design

Validation of feature The relationship between U-Value and thickness is an inversely proportional non-linear curve. Therefore, a similar curve is expected as the Pareto Optimal Front, after optimisation. From Figure 5.15, it can be seen that the the relationship between embodied energy and operating energy consumption is as expected.

5.3.5. EXPORT DESIGN TO OPERATE WITH THIRD-PARTY TOOLS

This test case aims at testing REQ-26 to REQ-28. Figure 5.16 illustrates the arrangement of the key components of *Embodied Optimisation Tool* and Geometry Gym's IFC components, to export the building design into IFC. Figure 5.17 illustrates a part of the Grasshopper script to export a design model to work with Karamba.

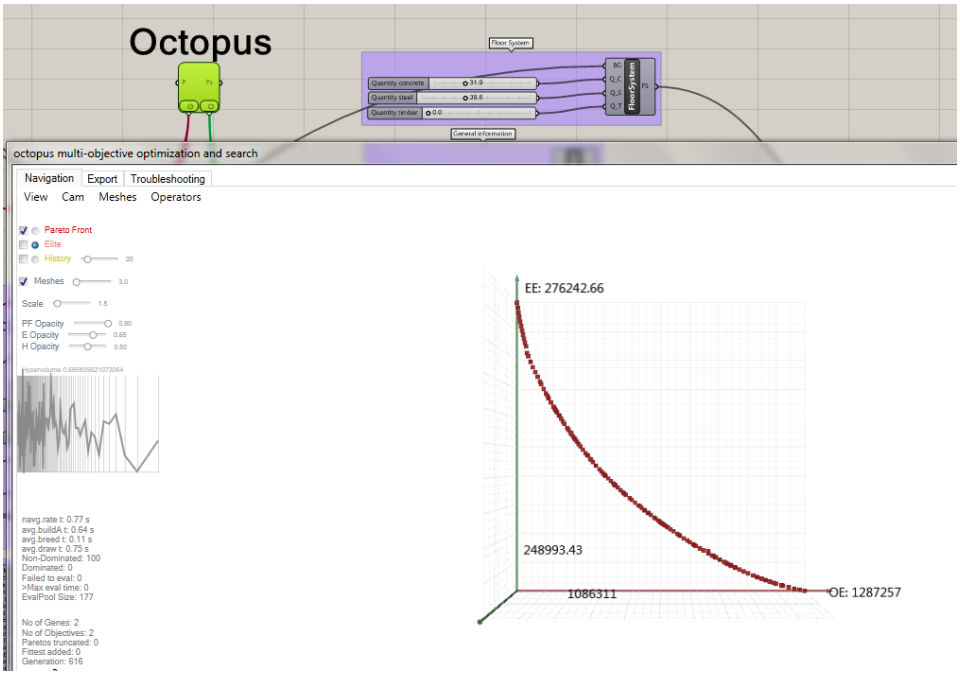

Figure 5.15: Optimisation Results from Octopus

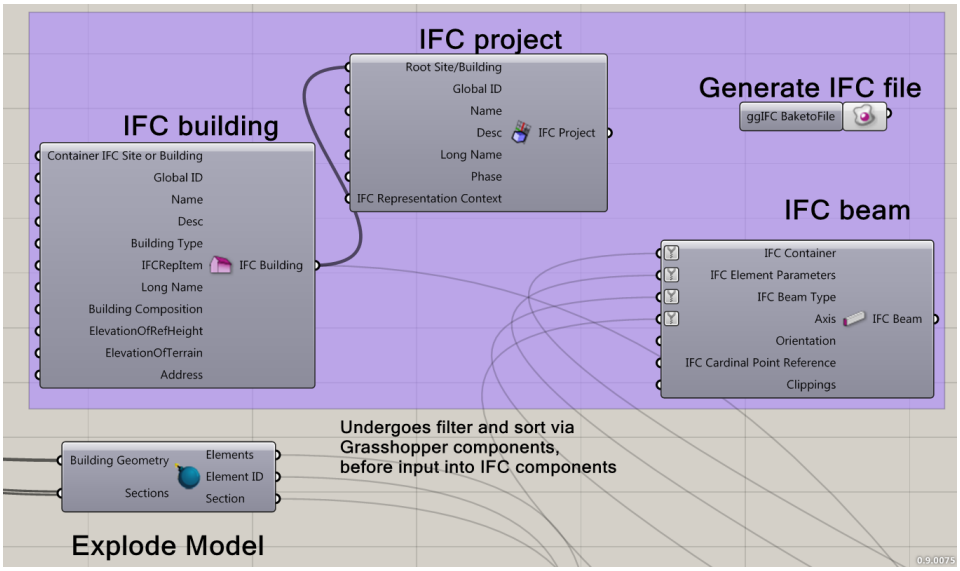

Figure 5.16: Test case to export design to IFC

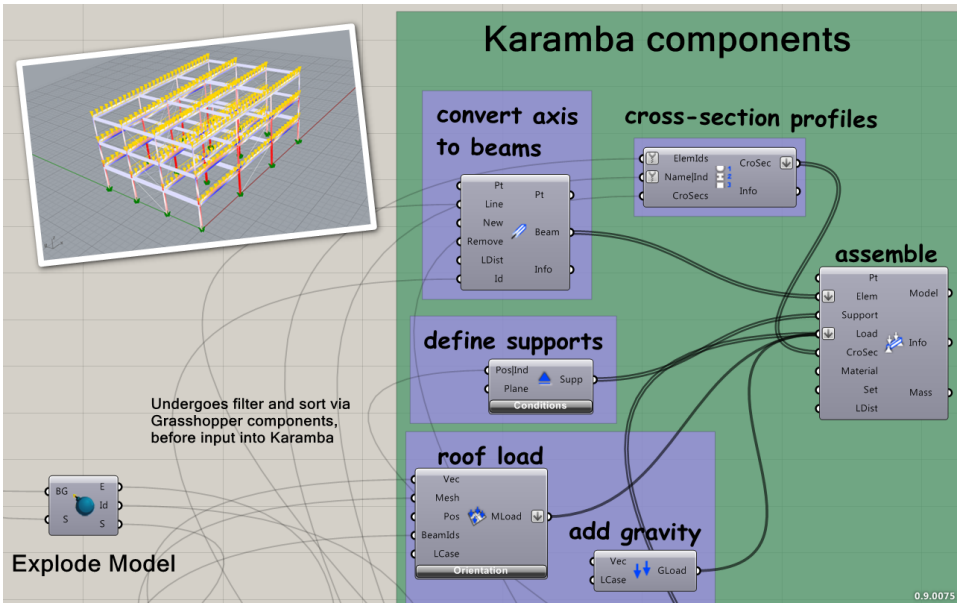

Figure 5.17: Test case to export design to Karamba

Description and Observations In both scripts, the common factor is the *Explode Model* component. This component uses the designed material profile sections and building geometry to provide a geometrical representation of each of the structural elements, together with a unique identifier and the name of the section profile. Further steps from this component consists of a combination of native Grasshopper components and Geometry Gym or Karamba components.

Requirements	Status	Remarks
REQ-26	Pass	-
REQ-27	Pass	Tested successfully with Geometry Gym IFC plug-in
REQ-28	Pass	Tested successfully with Karamba IFC plug-in

Table 5.7: Test case to export design

Validation of feature The IFC file created from the design is opened in Tekla BIMsight IFC model viewer and visually validated (Figure 5.18. The exploded design is analysed via Karamba components, with satisfactory results (as explained earlier under validation of Structural Analysis & Design).

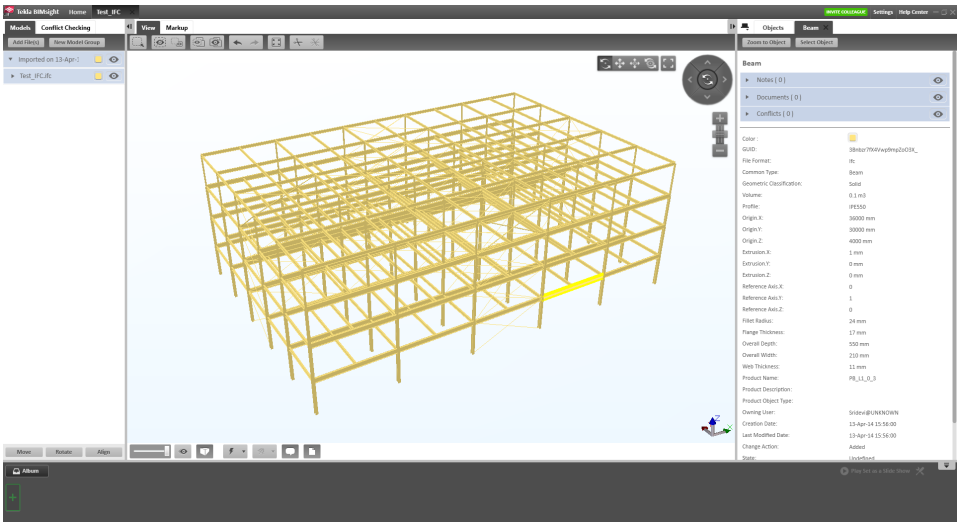

Figure 5.18: Exported design to IFC file, as opened in Tekla BIM-sight

6

DISCUSSION

The *Embodied Optimisation Tool* has been designed, developed, tested and validated in the development phase of the project. This chapter begins the final phase of the project, where the the tool is discussed from the perspective of achieving the project objectives and in regards to its limitations.

6.1. ACHIEVEMENT OF PROJECT OBJECTIVE

The main objective of this project was to

> **Develop a tool that supports decision making for the conceptual design stage, by assessing and optimising a building design so as to minimise its embodied energy and some aspects of the operational energy, depending on the adaptability required.**

During the research phase of this thesis, sufficient information was gathered on the different topics that define the above objective. Based on the knowledge gathered, a parametric computational framework for sustainable building design was developed. By applying this framework on the project objective, the objective was split into several functional requirements for the tool. A suitable design for the tool was then determined such that all these functional requirements are met with. The tool was then developed and the prototype tested and validated for the following features:

- Design minimum weight low rise rectangular grid office buildings in steel;
  - Calculate its embodied and operating energy consumption;
  - Determine its adaptability rating;
- Optimise the design so as to study the impact of structural grid, floor systems, façade panels (including photovoltaics) during the life of the building, etc., on the embodied energy consumption, operating energy consumption and adaptability of the building over its life-cycle.

To achieve these aforementioned possibilities, the tool works in both *forward manner* where the inputs are fixed, to provide the final design and energy calculations; or *backward manner* where some (or all) end results are fixed (desired) and using optimisation, the design values can be found to achieve these desired objectives. The observations made and results obtained from these test cases were satisfactory (Chapter 5). Therefore, it can be said that the *Embodied Optimisation Tool* succeeded in achieving the primary objective of this project.

Aside from the main objective, it was also aimed to answer the following questions during the course of this project:

- *What parameters define this tool and why?*
  The parameters defining this tool were identified partly during the research stage and partly during the design and development phase. These parameters and the associations between them were organised into the computational framework for sustainable building design, as illustrated in Figure 6.1.

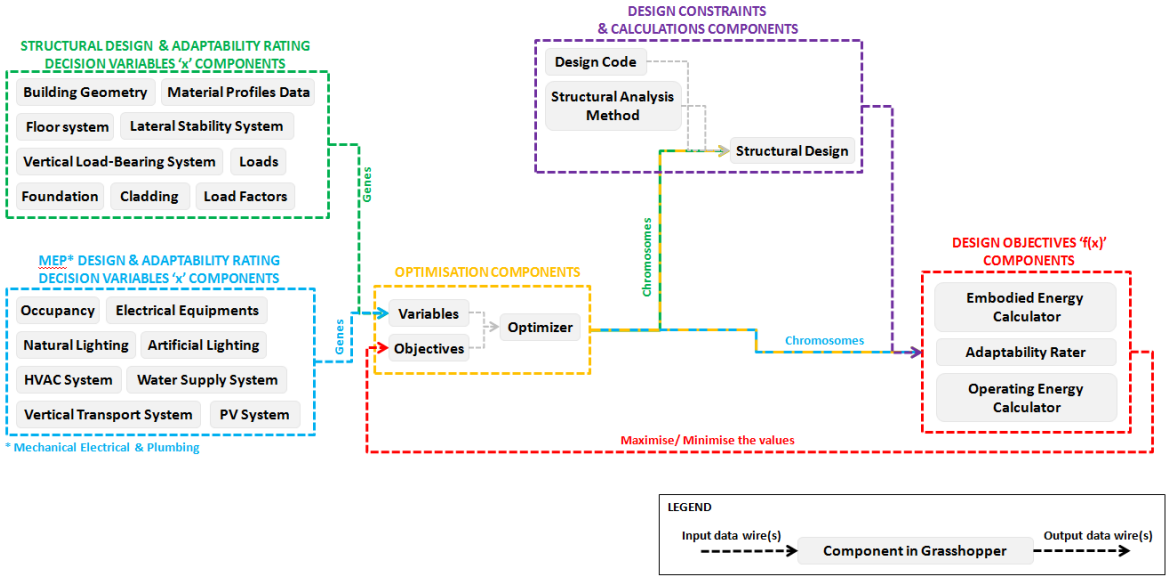

Figure 6.1: Different parameters in the tool and their associations

- *What challenges were overcome during the tool development?*
  Two of the biggest challenges faced during the tool development were in:
  - Making the computational framework, such that it satisfies the requirement specifications for the project. This required several iterations of improvement during the design and development stage;
  - Identifying and filtering out those parameters that have to be developed in detail so as to display the different applications of the tool, over other parameters.
- *What are the final limitations of the tool?*
  While the *Embodied Optimisation Tool* has been successful in achieving its primary objective, it is not without limitations. These limitations have been listed in the next section, together with suggestions for improvement.
- *What are the lines of possible future developments and explorations?*
  This question has been answered later in Chapter 8 - *Recommendations*.

6.2. LIMITATIONS OF TOOL FRAMEWORK & PROTOTYPE

Several functional and non-functional limitations were identified with the tool. The functional limitations or short-comings (per component) have been listed in Table 6.1. Possibilities for eliminating these drawbacks to further improving the tool have also been provided for each limitation. Non-functional limitations are just as important, they are categorized as follows:

- Risks
  - The design calculations in the current tool prototype are approximate, based on simple mechanics and hand calculations, and therefore, useful for early stages of design. For later stages, a well developed structural analysis and design software is recommended, until the design methods in the tool are improved to be at par with such software.
  - The tool was tested successfully with limited number of test cases. In depth testing is required with the assistance of end-users to identify and eliminate possible issues.
  - The current optimisation algorithms have been developed by third-parties and should therefore be used with caution. The user needs to be aware of possible pitfalls and have a back-up method or idea for verifying the results obtained, as observed during validation of results obtained with *Galapagos* in the previous chapter.

- Level of control and insight provided
  - Framework consists of independent functional parameters. By implementing the framework, the tool has also been simplified into independent components, giving user the control and insight into each parameter of framework. However, to prevent accidental tampering, the user is not provided with control over the contents of each components. Instead, calculation steps are made available by reports, which assist in verification.
  - While each individual component provides some information about its expected inputs and outputs, new users of Grasshopper will need some time to get used to the layout and working style of visual programming before they are able to work independently with the tool.
- Re-usability / flexibility / adaptability / customisability
  - The framework is very flexible, such that each of its interface is able to accommodate new or improved parameters. The tool is reasonably flexible, having implemented the framework. However, the current prototype of the tool cannot function independently of Rhino & Grasshopper.
- Interactivity / interoperability
  - The current prototype depends on the interoperability of Rhino & Grasshopper, with the aid of plug-ins like GeometryGym, Salamander, etc.

Component	**Limitations**	**Possibilities for Improvement**
Building Geometry	Only rectangular grid buildings	User is able to input 3D model/ actual shape of building
	Lateral stability only via diagonal wind bracing	Implement stability via other options, as well as their combinations
Loads	Only uniformly distributed area loads	User able to input different types of loads acting on structure
	Only permanent load, wind load, snow load	User is able to add other loads acting on structure
	Cannot override load combinations	User is able to override default load combination
	Location of application and direction of loads fixed	User is able to input the location and direction of loads acting on structure
Material	Only steel structures	Implement all construction materials
	Only I, O and [] shaped cross-section profiles	Implement all profiles
	Only .csv files	Implement other files types. Alternatively, tool accesses its own database, if no external file is provided
Analysis Method	Only simple braced frame analysis for rectangular grid buildings	Implement analysis components for all types of structures
	Method is based on simple mechanics and approximations	Refined analysis via finite element methods should be possible
Design Code	All checks from Eurocode 1993 are not implemented	Implement all design checks
	Only Steel Eurocode	Implement all building codes

Table 6.1: Possible improvements in tool prototype

Component	Limitations	Possibilities for Improvement
Designer	Designs only beams, columns, wind bracings	Implement design for all structural elements
	No information provided on horizontal or vertical clearance, façade grid and other adaptability related inputs	Implement calculation of this data, to provide as inputs for adaptability rating
Embodied Energy Calculator	Approximation (10%) for energy consumption during transportation and labour/construction	More accurate values should be determined
	Energy consumption calculated only for structural steel, flooring, glazing, PIR or stonewool insulation, photovoltaic	Identify when and where new or alternate material is used, include this in calculations
	Only .csv files	Implement other files types. Alternatively, tool accesses its own database, if no external file is provided
Operating Energy Calculator	Complicated layout, too many input components	Simplify the layout
	Calculation for HVAC fans and pumps not included	Include calculations for energy consumption of fans and pumps
	Only for rectangular grid office buildings	Implement calculations for all types of structures
Optimisation	Dependent on third-party developed components for multi-objective optimisation	Develop own component
Cumulative Energy Plotter	Slow algorithm, cannot be used during optimisation run	Improve the algorithm
Export to third-party tools	IFC file created using free-version of Geometry Gym	Develop own series of IFC related components
	Export tested only with free-version of Karamba	Develop own series of export components

Table 6.1: Possible improvements in tool prototype

7

CONCLUSIONS

In the previous chapter, it was concluded that the *Embodied Optimisation Tool* successfully achieved the project objective of optimising a building design so as to minimise its embodied energy and some aspects of the operational energy, depending on the adaptability required. This chapter sums up the conclusions drawn from this project.

- Most present day tools calculate (and sometimes, optimise) either embodied energy or operating energy of a building and does not consider adaptability at all. On the other hand, the *Embodied Optmisation Tool* is able to aid in studying the influence of these three aspects on the design of a building, from within a single tool. Thus, the tool is able to provide answers relating to sustainability, as early as in the conceptual design stage of a building.
- Since the tool aids in decision making from the conceptual stage of the building until its entire lifecylce, it can be concluded that both the framework and the tool are in accordance with the goals of Building Information Modeling.
- In the area of energy efficient building design, the tool is able to cater to the demands of the key players in the building industry - architects, engineers, building material and products manufacturers, as well as researchers, by aiding the user in:
  - Quickly designing a building, or importing/exporting a building design from/to an external software, for further actions;
  - Determining the life cycle embodied energy consumption and operating energy consumption of the particular building design;
  - Comparing and studying different building designs, materials or products to determine the most suitable (optimum) option for a given set of requirements or targets.
- Both the framework and the tool follows a very modular approach to building design, such that new features can be easily added or old features can be modified or replaced without affecting the functionality of the rest of the tool. This results in a flexible tool, which the user can adapt to satisfy his/her requirements. This implies that the tool has a lot of development potential to:
  - Easily move from crude to finer calculation methods with the progress of the design of the building;
  - Easily include other optimisation aspects such as cost assessment, or other design aspects such as fire engineering, to the list of tool features;
  - Not be limited by a particular building material or calculation method or optimisation algorithm, etc;
  - Design and optimise all types of structures, from buildings to offshore structures to bridges, from within a single tool.

  However, addition of new features or improvements to the current features require further study, which has been summarized in the next chapter - *Recommendations*.

8

RECOMMENDATIONS

The results obtained from this research project were found to be promising and opens up the possibilities for further study and development, both for the computational framework and the tool prototype. These recommendations have been outlined in this chapter.

- Several assumptions were made in the tool to determine the flexibility of the building (Tool 2010) and in calculating the life cycle embodied energy consumption. In particular, there is a lack of information on grey and induced energy consumptions of a building. Additional research is required to shed more light on these topics, so as to identify the stochastic factors involved in determining more accurate values for flexibility of the building and for complete life-cycle analysis.
- The limitations and short-comings of the current tool prototype were identified and listed in Chapter 6, together with suggestions for improvement. The tool requires further development to eliminate these short-comings. In order to implement these suggestions, further study is necessary to identify and understand the different types of structures and *better*:
  - Optimisation algorithms;
  - Analysis and design methods;
  - User interface and visualisations;
  - Energy calculations;
  - Database management system; and so on.
- In the current user interface of the tool (inside Grasshopper), each independent component provides detailed information about itself, however, the overall association of the different components is not visible to a new user, who is therefore unaware which combination of components are required to perform a particular task. Since there are a fixed set of components used in each task, it is recommended that a *wizard* or at the very least, an example file should be provided to guide new users on working with the tool.
- The tool functions from within the Rhino & Grasshopper environment and can interoperate with third-party developed plug-ins of Rhino. Modifications will be required to the different interfaces of the framework that are implemented in the tool, when it is desired that the tool is to function independently as a stand-alone software.
- User requirements are evolving by nature, based on the development of the tool itself and the development of other technologies in the same field. Therefore, improving the tool also requires extensive testing of its features and continuous feedback from users. Several software developers, including those of Rhinoceros and Grasshopper, participate heavily in discussions with the users regarding bugs or improvements. The increasing popularity of these software tools is an indication of the success of such interactions, and therefore, *Embodied Optmisation Tool* should not be an exception.
- Sustainability is and must be a target for everyone. By open sourcing this framework to the masses, we can work together in developing a universal standard for sustainability, for all.

BIBLIOGRAPHY

Allwood, J. M. & Cullen, J. M. (2012), *Sustainable Materials - With Both Eyes Open*, 1 edn, UIT Cambridge Ltd.

Brown, D. G., Iles, D. C. & Yandzio, E. (2009), *Steel building design*, 1 edn, The Steel Construction Institute.

Coello, C. A., Van Veldhuizen, D. A. & Lamont, G. B. (2002), *Evolutionary algorithms for solving multi-objective problems*, 1 edn, Kluwer Academic.

Drayton, P., Albahari, B. & Neward, T. (2003), *C# language pocket reference*, 1 edn, O'Reilly.

Environmental Protection Agency (2014), 'Basic information | green building |us epa', http://www.epa.gov/greenbuilding/pubs/about.htm. [Accessed 10-August-2014].

Foster, J. S. & Greeno, R. (2007), *Structure and fabric*, 1 edn, Pearson/Prentice Hall.

Huovila, P. (2007), *Buildings and climate change*, 1 edn, United Nations Environment Programme, Sustainable Consumption and Production Branch.

Institution of Structural Engineers (1999), *Building for a Sustainable Future: Construction Without Depletion*, 1 edn, Structural Engineers Trading Organisation Ltd.

International Institute for Environment & Development (2002), *Breaking New Ground*, 1 edn, Earthscan Publications Ltd.

Jones, D. L. (1998), *Architecture and the environment*, 1 edn, Overlook Press.

Jonkers, H. (2011), 'CT4100 Materials and Ecological Engineering - Lecture Notes'. Delft University of Technology.

Karamba3D (2014), 'Karamba3d', http://www.karamba3d.com/. [Accessed 10-August-2014].

Lebel, G. C. & Kane, H. (1987), *A guide to our common future*, 1 edn, Oxford University Press.

Milne, G. & Reardon, C. (2014), 'Embodied energy | Your Home', http://www.yourhome.gov.au/materials/embodied-energy. [Accessed 10-August-2014].

Ministerie van VROM (2014), *Criteria voor duurzaam inkopen van Nieuw te bouwen kantoorgebouwen*, AgentschapNL.

Mirtschin, J. (2014), 'Geometrygym', http://geometrygym.wordpress.com/. [Accessed 10-August-2014].

Perkins, C. & Tandler, J. (2005), *Embodied energy and carbon dioxide for typical office floor plates*, Arup.

Remoy, H. & van der Voordt, T. (2009), *Sunstainability by Adaptable and Functionally Neutral Buildings*, Delft University of Technology.

Robert McNeel & Associates (2014), 'Rhinoceros', http://www.rhino3d.com/. [Accessed 10-August-2014].

Rolvink, A. (2010), Structural components 2.0, Master's thesis, Delft University of Technology.

Rutten, D. (2011), 'Grasshopper discussions forum - galapagos multiple fitness', http://www.grasshopper3d.com/forum/topics/galapagos-multiple-fitness. [Accessed 10-August-2014].

Rutten, D. (2014), 'Galapagos - search results', http://ieatbugsforbreakfast.wordpress.com/?s=galapagos. [Accessed 10-August-2014].

Target Zero (2011), *Guidance on the design and construction of sustainable, low carbon office buildings*, Target Zero.

Tata Steel Construction (2014), 'Sustainability', http://www.tatasteelconstruction.com/en/reference/publications/sustainability_and_environment/. [Accessed 10-August-2014].

ter Mors, N. (2011), 'Bouwen voor leegstand', *HP/De Tijd* . [Accessed 10-August-2014].
URL: *http://www.hpdetijd.nl/2011-01-12/bouwen-voor-leegstand*

Tool, F. (2010), Ontwerptool voor de beoordeling van constructieve alternatieven op duurzaamheid, Master's thesis, Delft University of Technology.

Vierling, R. (2014), 'octopus | grasshopper', http://www.food4rhino.com/project/octopus. [Accessed 10-August-2014].

Wong, A. K. L. (2010), 'Embodied energy and carbon emissions: A case study of an infrastructure project in hong kong', *The HKIE Transactions* **17**(4).

Yalniz, O. (2008), *Sustainable Structural Design: Minimum Embodied Energy or Adaptability*, Arup.

Zitzler, E. (1999), Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, PhD thesis, Eidgenössische Technische Hochschule Zürich.

A

FLEXIBILITY RATING OF A BUILDING

This appendix provides the various graphs and tables utilized by the *Embodied Optimisation Tool* in determining the *Estimated Service Life (ESL) Factor*. Table A.1 summarises the various flexibility rating aspects involved.

Aspect	Depends on	Affects	ESL Factor
Stability system	Type if stability system	Freedom of layout	Refer Table A.2
Flexibility of voids	Type of flooring Applied reinforcement	Vertical transport	Refer Table A.3
	Support structure		
Load bearing capacity	Live load on floor	Job change (change in function) of building	Refer Figure A.1
	Location of floor (ground vs. upper)		-
Excess clearance	Vertical clearance (floor height) depends on: > Thickness of floor; > Space for installations (whether they are placed above, below or in the floor); > Floor function (ground vs upper).	Future developments (expansion, job change)	Refer Figure A.2
	Horizontal clearance (excess usable floor space)		Refer Figure A.3
	Overall factor per floor = sum of (factor * Gross Floor Area) / total Gross Floor Area of building		
Structural grid size	Grid distance of load bearing elements	Freedom of layout	Refer Figure A.4
	Grid size of façade which depends on: > Grid size of windows; > Installation and connection of interior walls on façade.	Maintenance of façade	Refer Figure A.5

Table A.1: Flexibility aspects and rating (Tool 2010)

Aspect	Depends on	Affects	ESL Factor
Façade	Wall grid	Changeability of walls	Refer Figure A.5
	Structural and building physics condition of façade		Refer Table A.4
	Architecture	-	-
Installations	Degree of accessibility of installations (hard/OK/easy)	Flexibility of installations	Refer Tables A.5 & A.6
	Positioning of pipe-zone and shaft (bad/ok/good)		

Table A.1: Flexibility aspects and rating (Tool 2010)

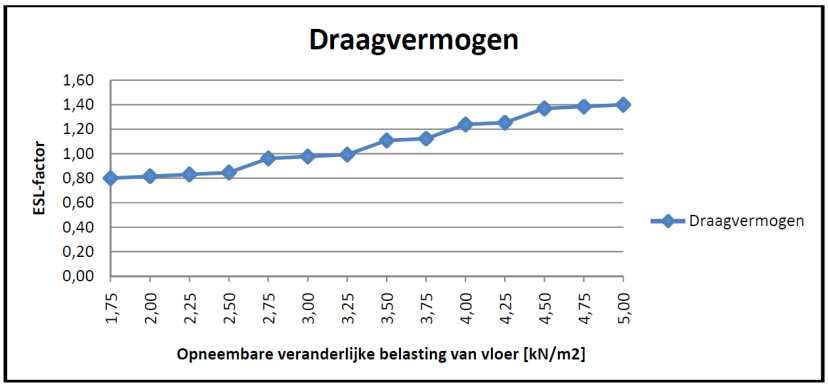

Figure A.1: Live load on floor - ESL factor (Tool 2010)

Stability through	ESL Factor
Core	1.00
Stability walls	0.90
Wind bracing	0.95
Portal frame	1.10

Table A.2: Stability system - ESL factor (Tool 2010)

Void area (sq. m)	Possibilities	ESL Factor
0 - 1	Use of small shafts possible	0.8
1 - 3	Use of small to bigger shafts possible	1.0
3 - 10	Use of elevator shaft (with risers) possible	1.1
> 10	Use of staircase possible	1.2

Table A.3: Flexibility of voids - ESL factor (Tool 2010)

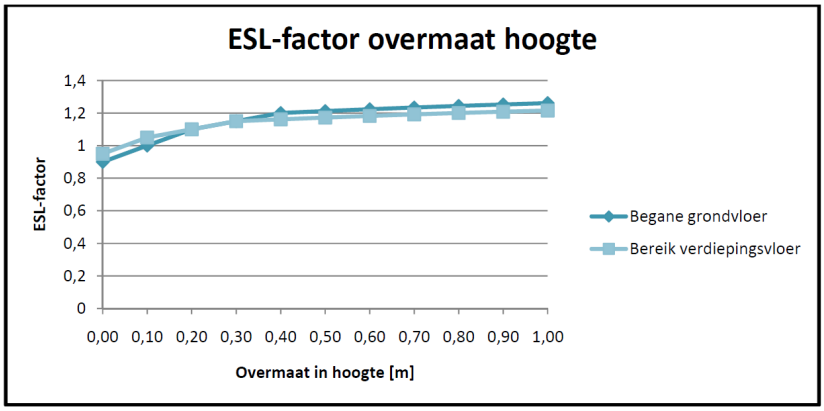

Figure A.2: Vertical clearance - ESL factor (Tool 2010)

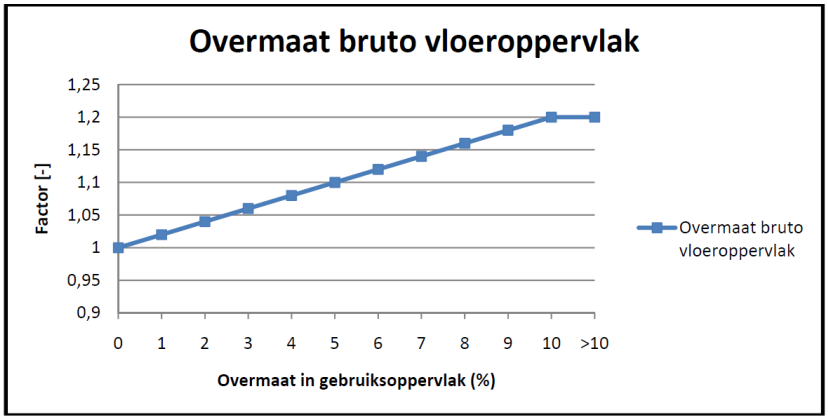

Figure A.3: Horizontal clearance - ESL factor (Tool 2010)

Façade wall	ESL Factor
Load bearing wall	0.7
Non-load bearing wall	1.4

Table A.4: Façade wall - ESL factor (Tool 2010)

Accessibility of installation facilities	Points
Easy access (built-level components)	4
Limited access (partly mounting and carrier level)	2
Poor access (carrier level components)	0

Positioning of pipe zone and shafts	Points
Good positioning (on the plot and central level)	4
Limited positioning (local)	2
Poor positioning (only at central level)	0

Table A.5: Installations - Points (Tool 2010)

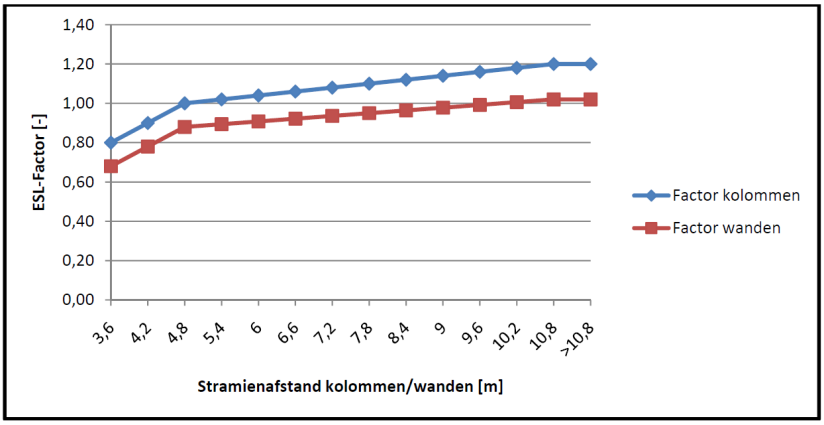

Figure A.4: Grid size (columns, walls) - ESL factor (Tool 2010)

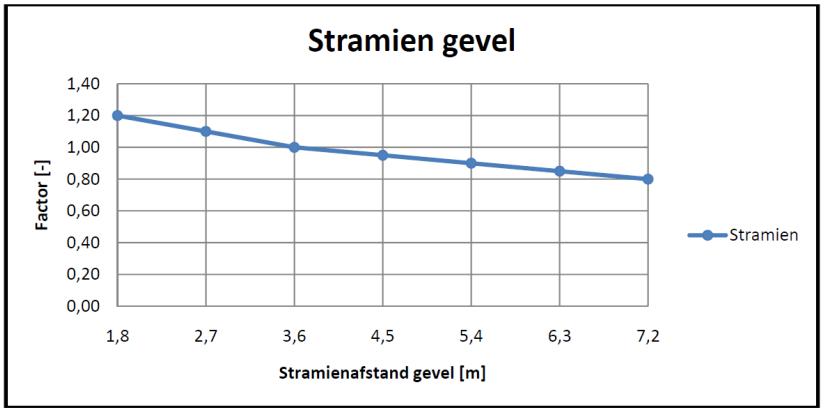

Figure A.5: Grid size (façades) - ESL factor (Tool 2010)

Total Points	ESL Factor
Minimum 0	0.8
Maximum 8	1.2
Interpolate the rest	

Table A.6: Installations - ESL factor (Tool 2010)

B

SOFTWARE INTERFACE

This appendix provides a brief look at the software interface between the user and the *Embodied Optimisation Tool*.

B.1. RHINOCEROS

Rhinoceros (Rhino) is a 3-D modeling tool, developed by Robert McNeel & Associates. It specializes in free-form non-uniform rational B-spline (NURBS) modeling. Aside from NURBS curves, Rhino works with surfaces, solids (with no limits on complexity, degree, or size), and supports polygon meshes and point clouds. It contains several tools for unrestricted editing, analyzing, documenting, rendering, animating and translating NURBS objects, thereby supporting drafting, engineering, analysis and manufacturing large or small models. It helps to develop designs quickly and accurately and be easily communicated across the various participants of a project, with the help of the following features:

- It has multi-disciplinary functions and is used in building architecture, naval architecture, jewelry design, etc.;
- It requires a short learning curve;
- It has a relatively low cost with no maintenance fees;
- It has worldwide support;
- It no special hardware requirements;
- It can read and repair extremely challenging mesh files;
- It supports 3D digitizing arms, scanners and printers;
- It is highly compatible and can import and export over 30 file formats, thus supporting interoperability between various programs in the field of design.

Apart from these benefits, Rhino is also customizable. Plug-ins for Rhino have been developed my McNeel & Associates as well as over 100 third-party plug-ins are available to suit specialized functions. To develop custom plug-ins, Rhino also provides scripting languages based on Visual Basic, C#, C++ and more recently Python, and software development kit (SDK) for .NET and C++ (Robert McNeel & Associates 2014).

B.2. GRASSHOPPER

With development of the .NET plug-in Grasshopper, Rhino has become quite popular in the field of architecture and civil engineering. Grasshopper is a graphical algorithm editor tightly integrated with Rhino's 3-D modeling tools. It is developed by David Rutten at Robert McNeel & Associates. It can be called a visual programming language, since it creates programs by dragging components onto a canvas and connecting outputs of components with inputs of other components. Since there is no need to learn scripting, it is a popular parametric modeling tool and is used by students and professionals alike, to build generative algorithms and explore new designs and forms. An SDK is also available for .NET to develop custom plug-ins for Grasshopper in Visual Basic or C#.

B.3. DEPENDENCIES

Below is a list of libraries that Embodied Optimisation Tool depends on for its various functions:

- `RhinoCommon.dll`
  This is the Rhinoceros .NET plugin software development kit (SDK). It provides the application programming interface (API) for integration with Rhinoceros.
- `Grasshopper.dll`, `GH_IO.dll`
  These libraries are part of the Grasshopper SDK and provide an API for accessing and modifying the Grasshopper components library.
- `itextsharp.dll`
  This library aids in generating portable document format (pdf) of the calculations performed in the tool, for verification by the user.

C

List of Components in Embodied Optimisation Tool

The various components available in *Embodied Optimisation Tool 1.0* are listed below:

- Analysis & Design Components
  - Designer;
  - Simple Braced Frame Analysis;
  - Eurocode 1993.
- Calculators Components
  - Adaptability Rater;
  - Embodied Energy Calculator;
  - Operating Energy Calculator;
  - Eurocode Reduction Factors Calculator.
- Design Data Components
  - Building Geometry;
  - Façade;
  - Floor System;
  - General Information;
  - Load Factors;
  - Loads;
  - Steel Profiles;
  - Electrical Equipments;
  - HVAC Efficiencies;
  - HVAC Systems;
  - HVAC Temperature Pressure;
  - Installations;
  - Lighting;
  - Occupancy;
  - Photovoltaics;
  - Thermal Comfort;

  - Vertical Transport System;
  - Water Supply System;
  - Weather Data.
- Utilities Components
  - Explode Model;
  - Model From Karamba;
  - Cumulative Energy Plotter;
  - Report Generator.

The above mentioned components are what ultimately interact with the user (through Grasshopper) and therefore, follows certain common traits with every component of Grasshopper (and its plug-ins). These traits are obtained from implementing the parent or base class `GH_Component`. Classes, simply put, are derived data types in C#, with its own fields, properties and behaviours. Figure C.1 is a class diagram for `AdaptabilityRaterComponent`, illustrating the various traits that were inherited from the base class. All the classes on its right (which have been minimized) contain identical properties and behaviors, except for the name of the constructor. A brief explanation of these *traits* are as follows:

- Properties
  - `ComponentGuid` - Assigns a unique identifier to the component;
  - `Exposure` - Determines whether the component is hidden from the user or visible as primary, secondary, etc. in its category;
  - `Icon` - Assigns an icon for the component.
- Methods
  - `Constructor` - Initialises the instance of the class, whenever the user accesses the component. The constructor has the same name as the class itself;
  - `RegisterInputParams` - Registers the input parameters of the component, including their names and descriptions;
  - `RegisterOutputParams` - Registers the output parameters of the component, including their names and descriptions;
  - `SolveInstance` - Contains all tasks/actions performed by the component, whether it is simply wrapping together the input design data into a single output parameter for further use, or calculations to obtain energy values, etc.

However, it is to be noted that not all the components are included in this Figure C.1. This is because some of the components contain additional fields or behaviors which may or may not be implemented from the parent class. Separate class diagrams are provided for these classes.

Figure C.2 displays the class diagram for `FaçadeComponent`, which implements an additional method of base class called `AppendMenuItems`. This method is used to provide the user with multiple choice options to be selected by right clicking the component. The additional fields are also used to determine the option selected by the user. The minimized classes on the right also contain similar class diagrams.

The class `Eurocode1993Component` uses an additional method `AscendingOrderCheck` to ensure that the yield strengths are in the ascending order (Figure C.3). `SteelProfilesComponent` use a sorting algorithm `CompareProfilesByMass` to order the various section profiles by increasing mass (Figure C.4).

Behind the scenes, several other classes and interfaces are in play, for a variety of purposes. All the Design Data components ties together all the input data into a custom data type, for further use. The process of creating custom data types in Grasshopper involves the implementation of two base classes `GH_Goo<>` and `GH_Param<>`. Figure C.5 depicts the class diagrams for classes `Analysis` and `AnalysisParam`, which implements the respective base classes. Other minimized classes have identical class diagrams.

The innermost layer of classes in the tool are composed of unique fields which are private to the particular class and therefore is accessible only to the methods of that class. The methods are public, by which other

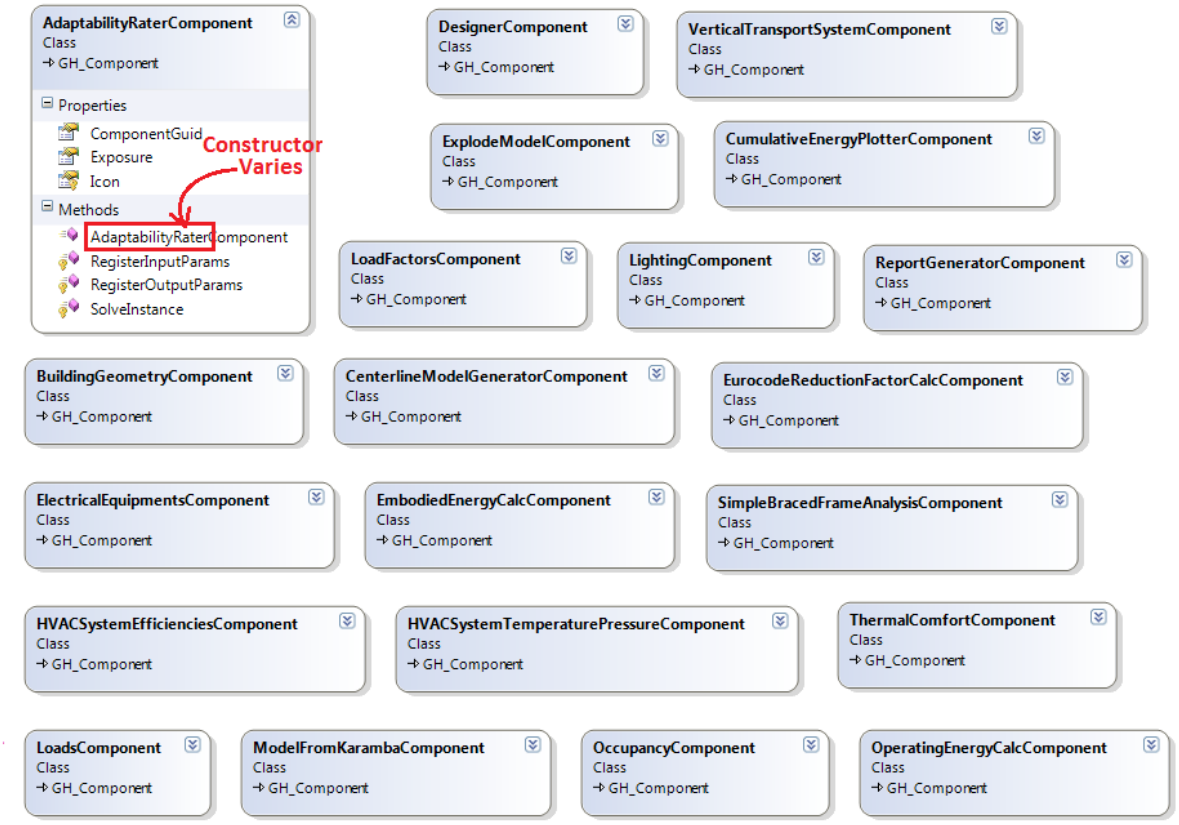

Figure C.1: Class Diagram for Embodied Optimization Tool Components - Type 1

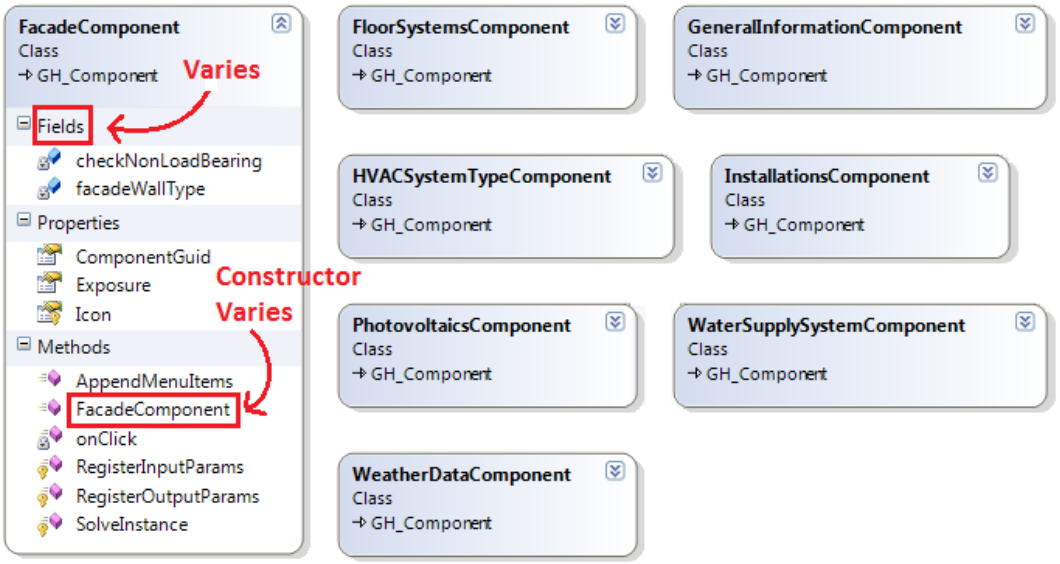

Figure C.2: Class Diagram for Embodied Optimization Tool Components - Type 2

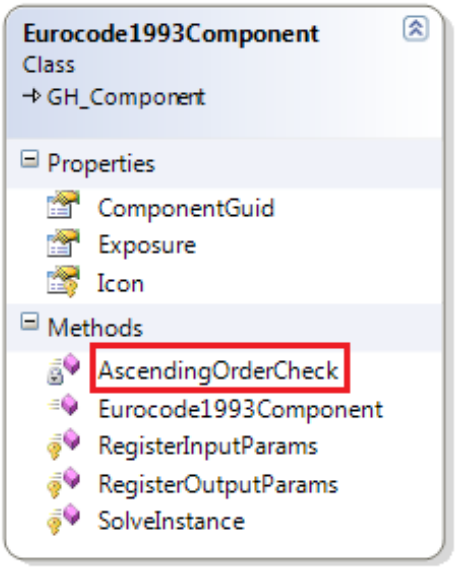

Figure C.3: Class Diagram for Embodied Optimization Tool Components - Eurocode 1993

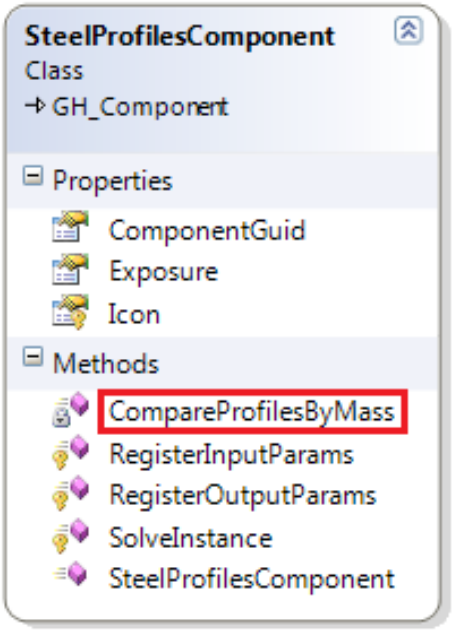

Figure C.4: Class Diagram for Embodied Optimization Tool Components - Steel Profiles

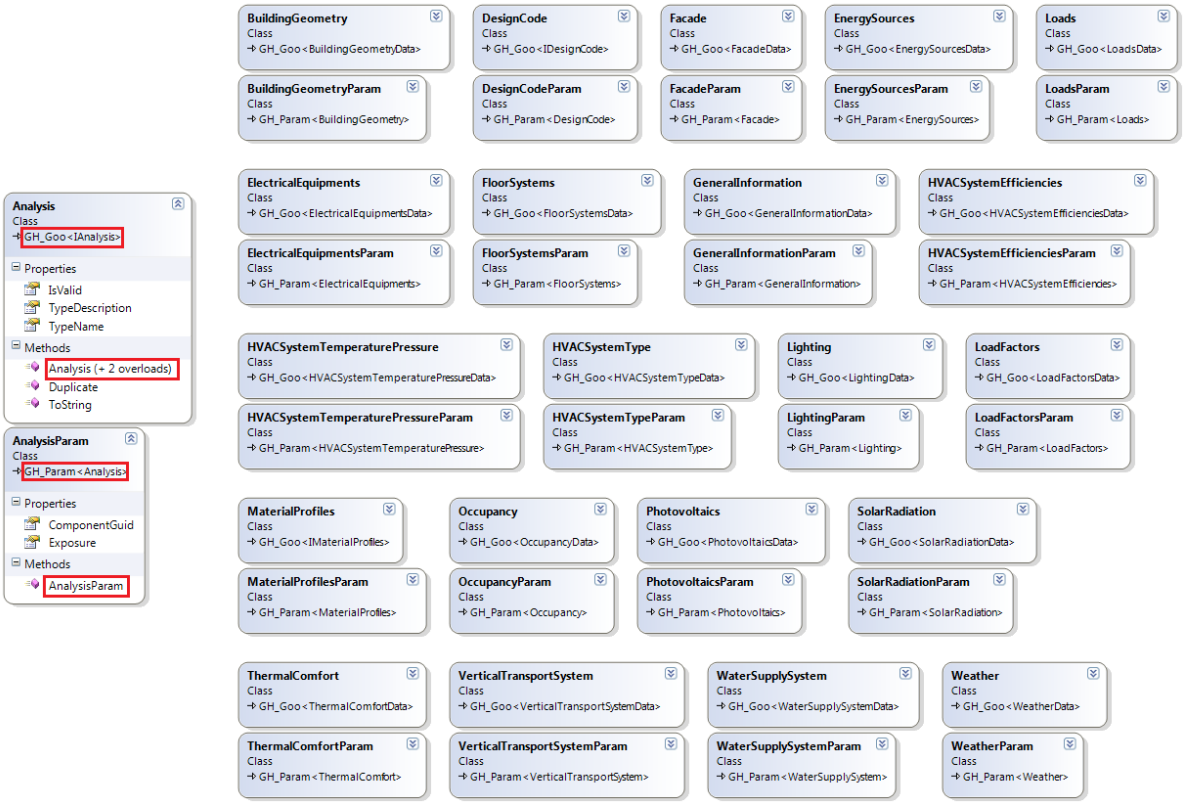

Figure C.5: Class Diagram for Embodied Optimization Tool - Custom Parameters

classes are able to access the information inside the fields, without directly accessing the fields itself. Therefore, for each field, a method was created to provide the particular information. Aside from these methods, there are also methods that perfom certain tasks or calculations, etc. using the various fields and providing a result to other classes.

Some of these methods are also created by implementing a parent interface. Such interfaces are created to ensure that all the classes that fall in its family must display certain common behaviours. Keeping with the parametric computational framework for sustainable building design (Figure fig:Framework), these interfaces ensure that new or additional design variables or objectives, improved calculation methods or optimisation algorithms can be easily incorporated into the tool. The *Embodied Optimisation Tool* currently contains three interfaces - `IAnalysis`, `IDesignCode` and `IMaterialProfiles`. The class `SimpleBracedFrameAnalysis` implements `IAnalysis` (Figure C.6), class `Eurocode1993` implements `IDesignCode` (Figure C.7) and class `SteelProfilesData` implements `IMaterialProfiles` (Figure C.8).

Each design data category is unique in terms of the information it contains. Therefore, it is not useful to create a universal interface like `IDesignData` for all design data categories, in a manner similar to analysis methods or design codes. However, it should be possible to have individual interfaces for each category, such as `IBuildingGeometryData`, `IElectricalData`, etc. if necessary.

All design data categories in this project implement the base classes from Grasshopper (as explained earlier in this chapter). New design data can be easily added to the tool, by implementing the exact same classes. Hence, these base classes, while officially not interfaces, act in a similar manner to unify the design data. Class diagrams were also drawn for these design data:

- Building Geometry Data (Figure C.9);
- Electrical Equipments Data (Figure C.10);
- Façade Data (Figure C.11);
- Floor Systems Data (Figure C.12);

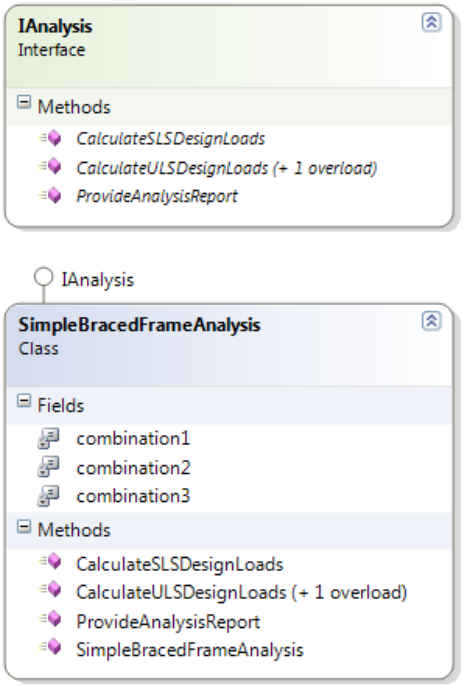

Figure C.6: Class Diagram for Embodied Optimization Tool - interface IAnalysis & class SimpleBracedFrameAnalysis

- General Information Data (Figure C.13);
- Water Supply System Data (Figure C.14);
- HVAC System Data (Figure C.15);
- Lighting Data (Figure C.16);
- Loads and Load Factors Data (Figure C.17);
- Occupancy Data (Figure C.18);
- Photovoltaics Data (Figure C.19);
- Vertical Transport System Data (Figure C.20);
- Solar Radiation Data (Figure C.21);
- Thermal Comfort Data (Figure C.22);
- Weather Data (Figure C.23);

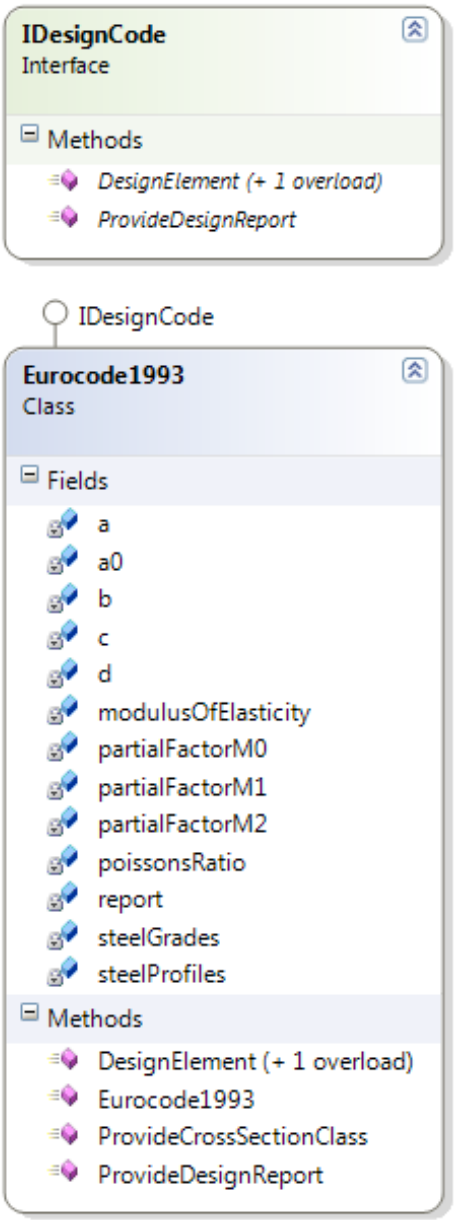

Figure C.7: Class Diagram for Embodied Optimization Tool - interface IDesignCode & class Eurocode1993

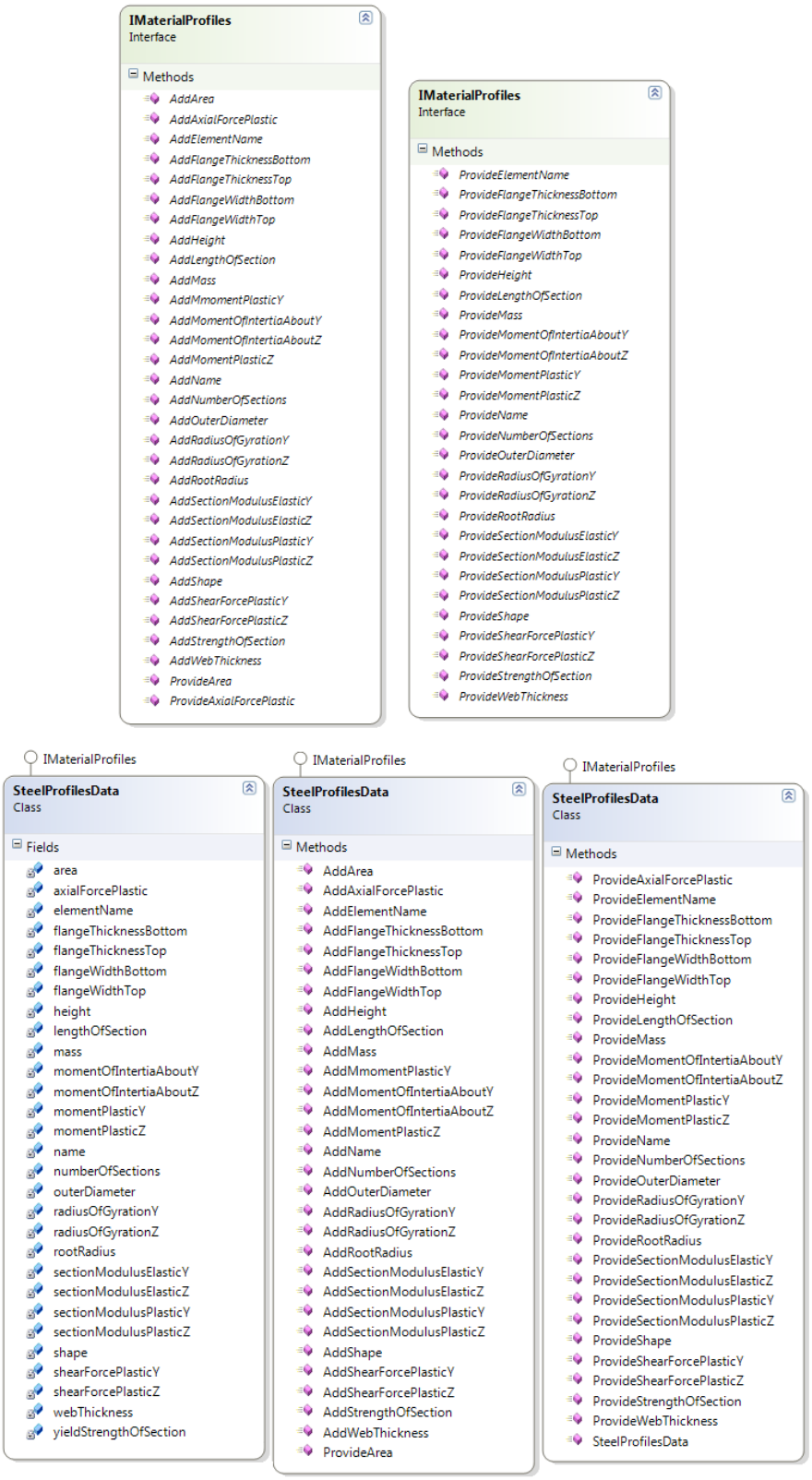

Figure C.8: Class Diagram for Embodied Optimization Tool - interface IMaterialProfiles & class SteelProfilesData

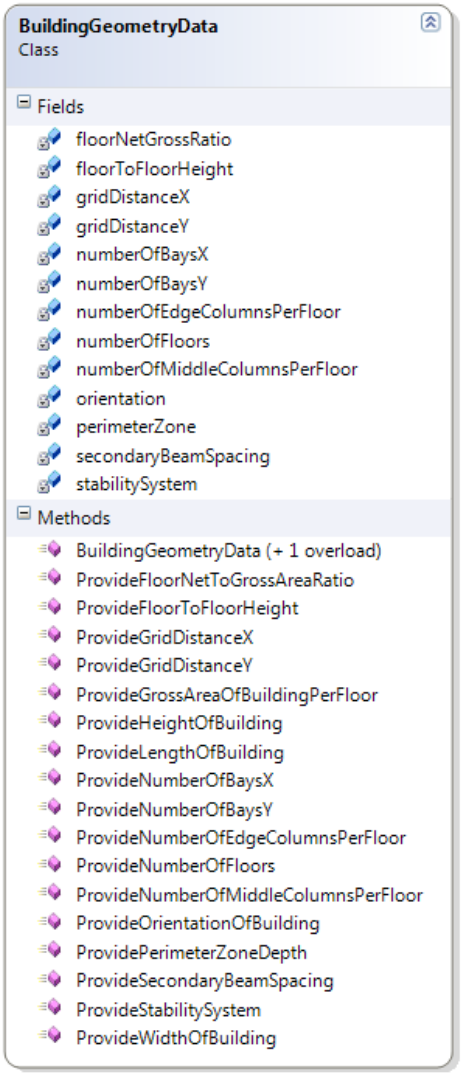

Figure C.9: Class Diagram for Embodied Optimization Tool - Building geometry data

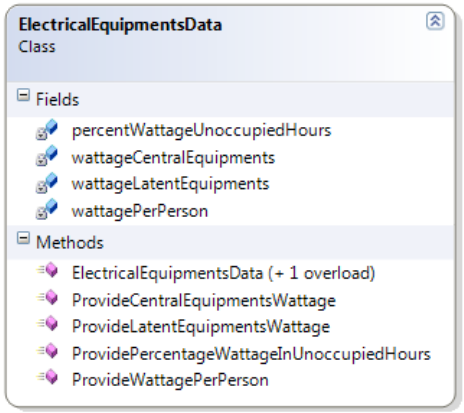

Figure C.10: Class Diagram for Embodied Optimization Tool - Electrical equipments data

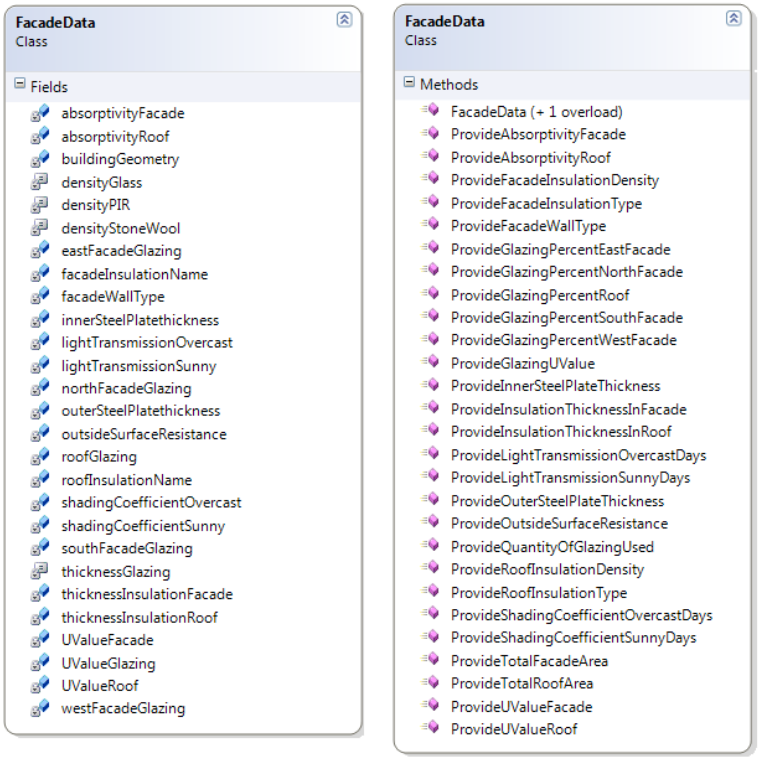

Figure C.11: Class Diagram for Embodied Optimization Tool - Façade data

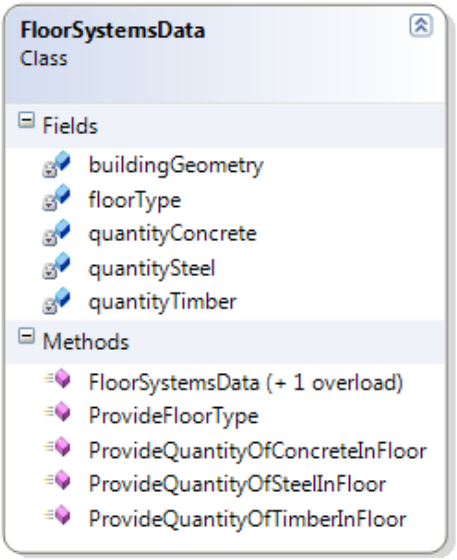

Figure C.12: Class Diagram for Embodied Optimization Tool - Floor systems data

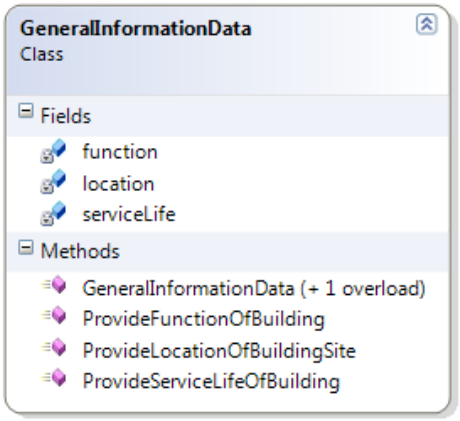

Figure C.13: Class Diagram for Embodied Optimization Tool - General information data

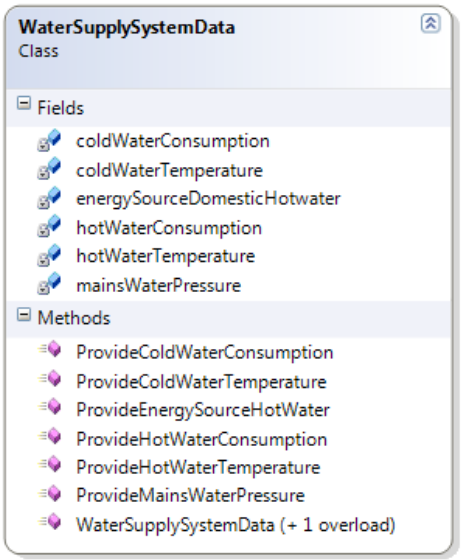

Figure C.14: Class Diagram for Embodied Optimization Tool - Water supply system data

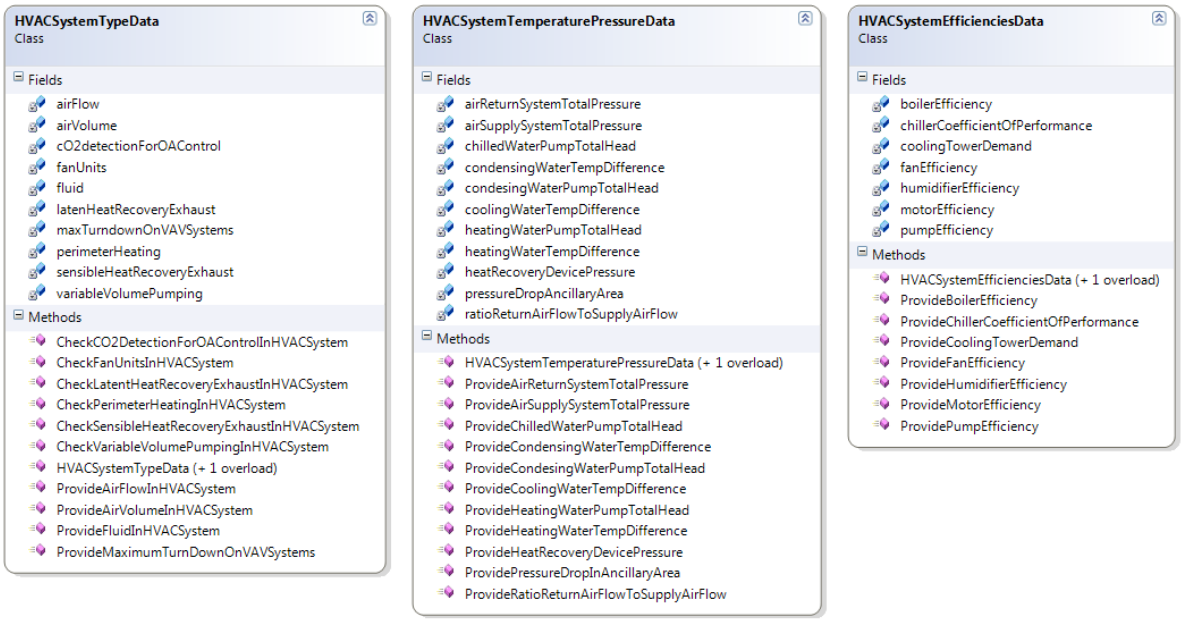

Figure C.15: Class Diagram for Embodied Optimization Tool - HVAC systems data

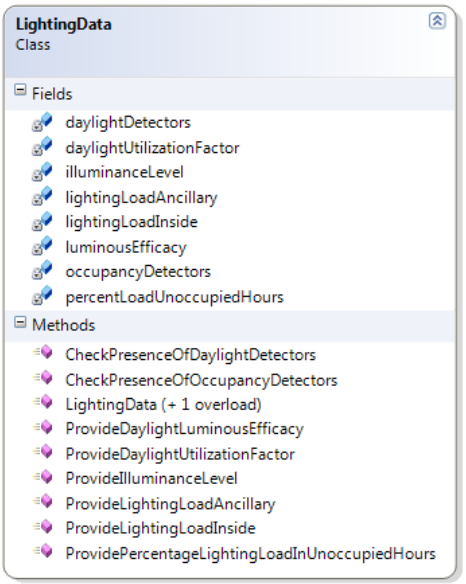

Figure C.16: Class Diagram for Embodied Optimization Tool - Lighting data

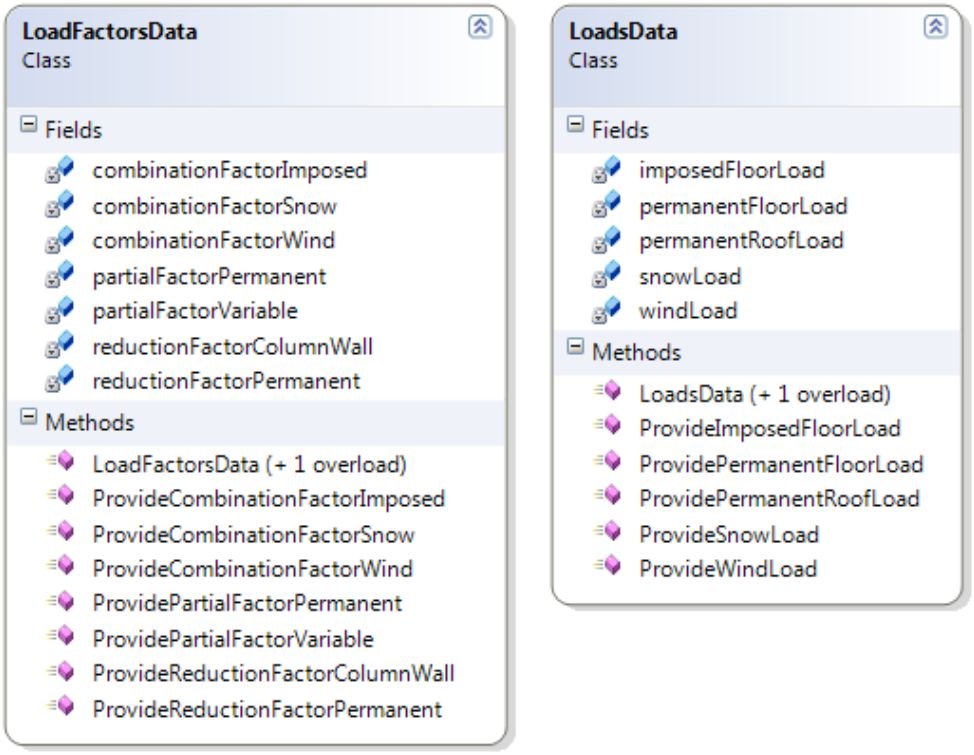

Figure C.17: Class Diagram for Embodied Optimization Tool - Loads and load factors data

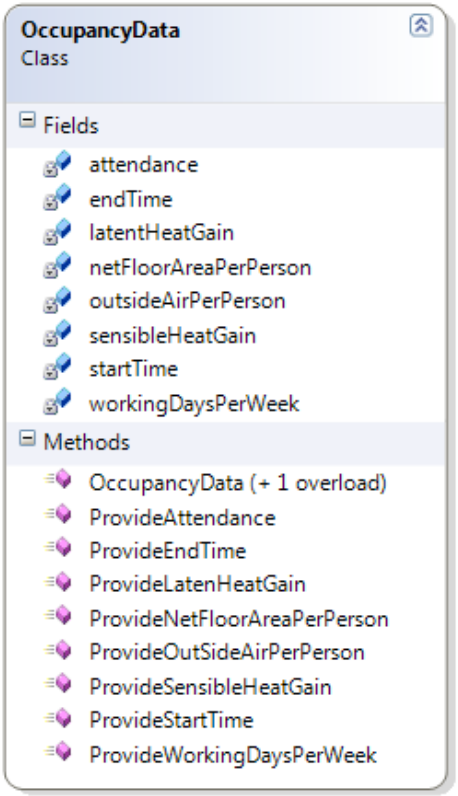

Figure C.18: Class Diagram for Embodied Optimization Tool - Occupancy data

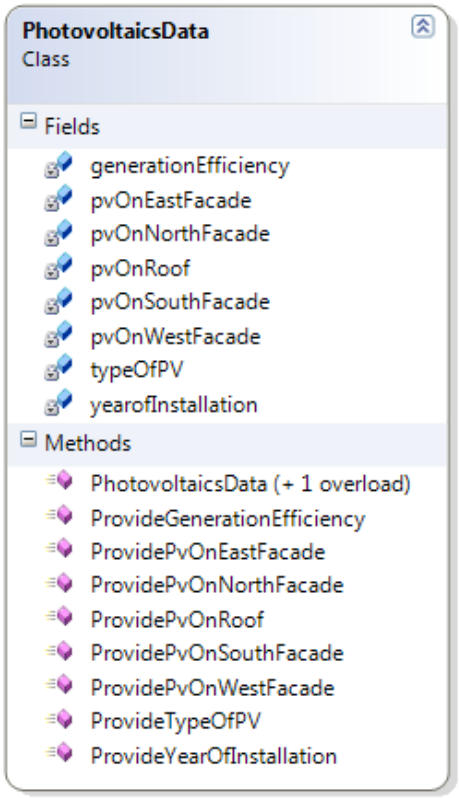

Figure C.19: Class Diagram for Embodied Optimization Tool - Photovoltaics data

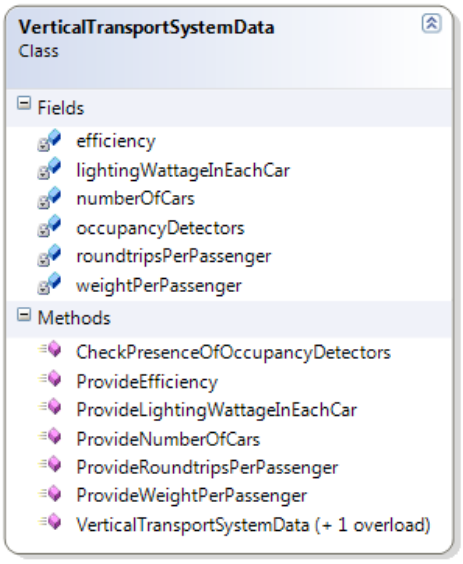

Figure C.20: Class Diagram for Embodied Optimization Tool - Vertical transport system data

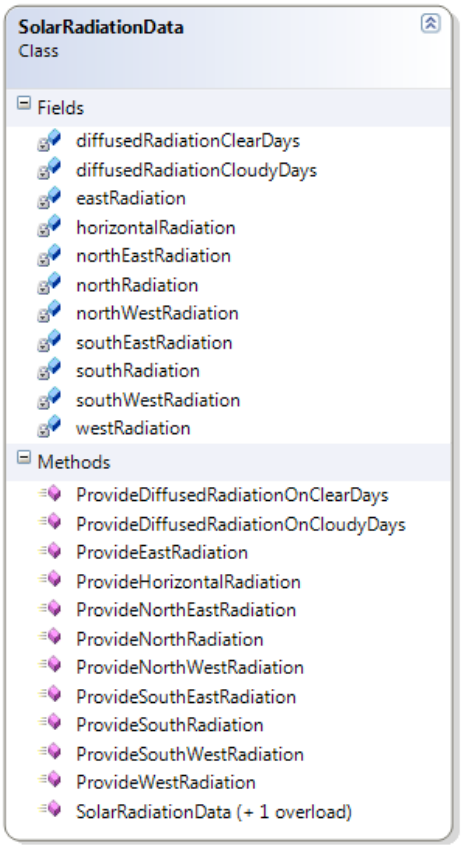

Figure C.21: Class Diagram for Embodied Optimization Tool - Solar radiation data

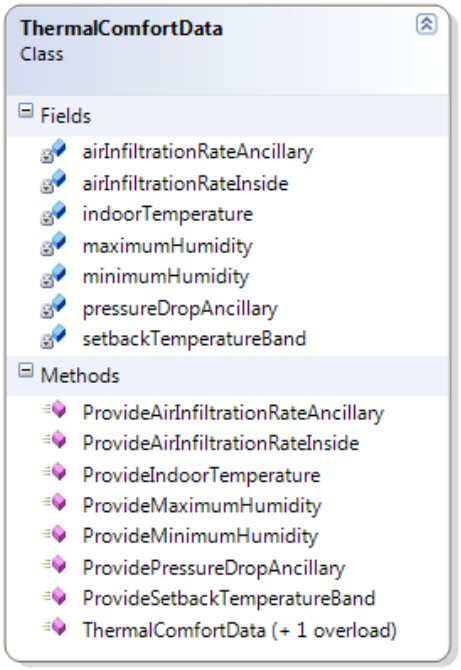

Figure C.22: Class Diagram for Embodied Optimization Tool - Thermal comfort data

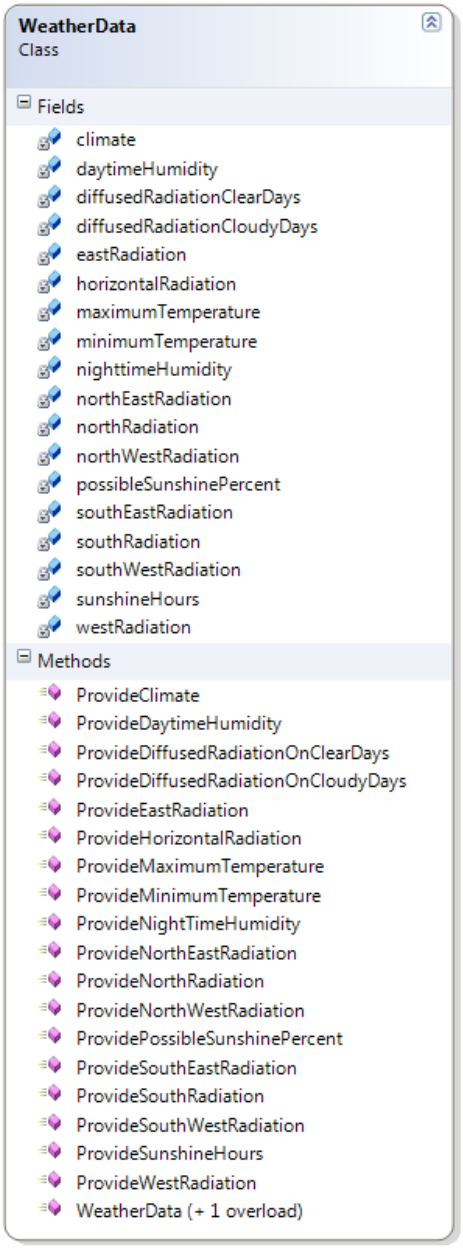

Figure C.23: Class Diagram for Embodied Optimization Tool - Weather data

D

STEEL BUILDING DESIGN

This appendix provides a quick glimpse (in the next pages) at the design example for a four storey braced frame in steel, from The Steel Construction Institute. For the complete example, please refer to Brown et al. (2009). This example was used in validating the design calculations performed within the *Embodied Optimisation Tool*, This is followed by the calculations report obtained from the tool, when the design data from the example was used as the input.

Silwood Park, Ascot, Berks SL5 7QN
Telephone: (01344) 636525
Fax: (01344) 636570

CALCULATION SHEET

Job No.	CDS 165	Sheet	1 of 17	Rev	A
Job Title	Multi-storey braced frame				
Subject	Sway stability and bracing design				
Client	SCI	Made by	EDY	Date	Feb 2009
		Checked by	DGB	Date	May 2009

Appendix A Sway stability of a braced frame

A.1 Introduction

The frame of an office building is shown in Figure A.1; it consists of steel beams and columns arranged on a 7 m × 7 m grid. The frame has been designed on the basis of "Simple Construction" - i.e. pinned connections between beams and columns. Resistance to sway is provided on each 49 m long side by two 3.5 m braced bays, as shown in Figure A.1; bracing is also provided parallel to the 28 m side, but this is not considered in this example. Typically these braced frames are positioned around stair or lift cores and so it has been assumed that the width of the braced bays is 3.5m.

These calculations demonstrate:

- The calculation of design values of actions (loads).
- The use of the reduction factor for axial forces on the columns of multi-storey buildings.
- The derivation of equivalent horizontal forces, representing the effects of sway imperfections in the frames.
- A first-order analysis of the braced frames to determine the forces in the bracing system and the sway stiffness of the frames.
- The conclusion that second-order effects need to be taken into account in this example and calculation of the amplification factor to be applied.
- The design of the diagonal bracing.
- The consideration of imperfection forces due to splices and restraint forces in the bracing system.

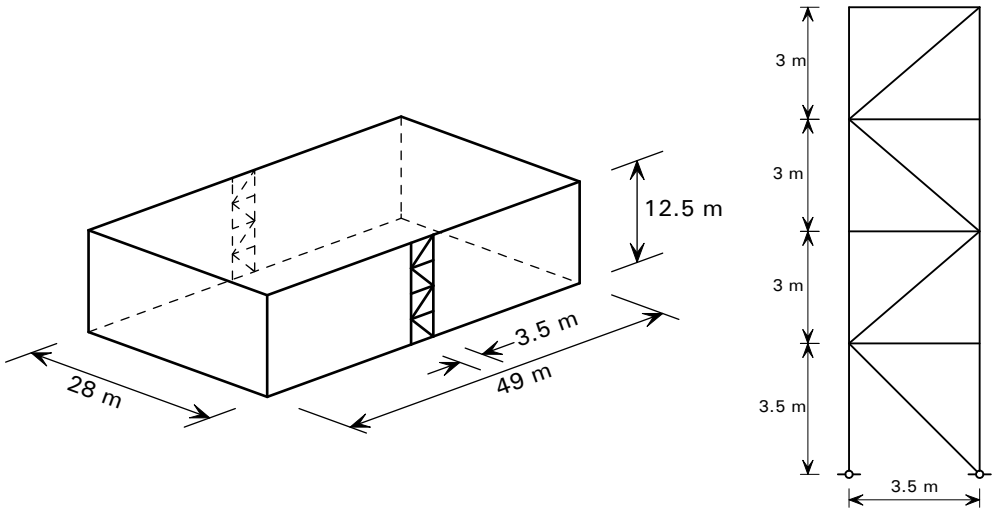

Figure A.1 *Building dimensions and braced frame*

Design of bracing system in a multi-storey braced frame	Sheet 2 of 17	Rev A

A.2 Actions

Vertical loads on roof and floors

The characteristic uniformly distributed loads on the roof and each floor, as agreed with the client are:

Roof:

Permanent action $g_{k,r}$ = 3.5 kN/m^2 (includes self weight of beams and columns)
Variable action $q_{k,r}$ = 1.0 kN/m^2

Floors:

Permanent action $g_{k,f}$ = 3.5 kN/m^2 (includes self weight of beams and columns
Variable action $q_{k,f}$ = 6.0 kN/m^2

Horizontal loads

For the location and local topography of the building and its shape, the following parameters have been determined:

Wind pressure = 0.8 kN/m^2
Overall wind force coefficient = 1.1

The projected area of the vertical face of building, for the bracing under consideration = 28 × 12.5 = 350 m^2

Characteristic value of total wind load on the building face
= 0.8 × 1.1 × 350.0 = 308 kN

There are two braced frames resisting horizontal loads acting on the face under consideration, therefore the total wind load per braced frame
= 308 ÷ 2 = 154 kN

A.3 Factors on actions

For the design of structural members not involving geotechnical actions the partial factors for actions to be used for ultimate limit state design should be obtained from the National Annex to BS EN 1990. From NA.2.2.3.2 and Table NA.A1.2:

BS EN 1990 A1.3.1(4)

Partial factors:

Permanent actions (Unfavourable)	$\gamma_{Gj,sup}$	= 1.35
Reduction factor for unfavourable permanent actions (6.10b)	ξ	= 0.925
Variable actions (Unfavourable)	$\gamma_{Q,1}$	= 1.50
(Imposed load on floors and wind load)	$\gamma_{Q,i}$	= 1.50

BS EN 1990 Table NA.A1.2(B)

Factors on accompanying actions:

Imposed loads on buildings - Category B: Office areas	ψ_0	= 0.7
Wind loads on buildings	ψ_0	= 0.5
Snow loads (Altitude < 1000 m above sea level)	ψ_0	= 0.5

BS EN 1990 Table NA.A1.1

Note that for favourable actions:

Permanent actions (Favourable)	$\gamma_{Gj,inf}$	= 1.00
Variable actions (Favourable)	$\gamma_{Q,1}$	= 0.0
Imposed load on floors and wind load	$\gamma_{Q,i}$	= 0.0

A.4 Combination of actions for ultimate limit state (ULS)

BS EN 1990 presents two options for determining the combination of actions to be used for the ultimate limit state. The options are to use expression (6.10) or to determine the less favourable of expressions (6.10a) and (6.10b). The National Annex to BS EN 1990 allows the designer to make the choice. In this example, expressions (6.10a) and (6.10b) are considered. In practice, expression (6.10b) will often be critical.

When considering the possible combinations in accordance with expressions (6.10a), one action is identified as the "main variable action", which must be considered in combination with all "accompanying variable actions".
Similarly, when considering expression (6.10b), one action is identified as the "leading variable action", which must be considered in combination with all "accompanying variable actions".

Expressions (6.10a) and (6.10b) are shown below. In this example there are no pre-stressing actions hence $P = 0$.

$$\sum_{j\geq 1}\gamma_{G,j}G_{k,j}\ "+"\ \gamma_P P\ "+"\ \gamma_{Q,1}\psi_{0,1}Q_{k,1}\ "+"\ \sum_{i>1}\gamma_{Q,i}\psi_{0,i}Q_{k,i} \qquad (6.10a)$$

$$\sum_{j\geq 1}\xi_j\gamma_{G,j}G_{k,j}\ "+"\ \gamma_P P\ "+"\ \gamma_{Q,1}Q_{k,1}\ "+"\ \sum_{i>1}\gamma_{Q,i}\psi_{0,i}Q_{k,i} \qquad (6.10b)$$

Clause A1.2.1 Note 1 allows the combination of actions for building design to be based on not more than two variable actions, although the application of this clause is a matter of engineering judgement. In this example, the variable action on the floors and the variable action on the roof have been considered to act simultaneously, which is conservative. The wind action is taken as the second variable action.

In this example, the combinations that will be considered are therefore:

Combination 1: Permanent loads, floor loads, roof loads and wind

Combination 2: Permanent loads, floor loads, roof loads and wind

The "leading" or "main" variable action is underlined, and is hereafter referred to simply as the "leading" action

A.5 Design values of actions

Combination 1 – variable floor and roof loads as "leading" actions to 6.10a

Substituting the values of the factors on actions, (6.10a) becomes:

Roof loads

$$1.35G\ "+"\ 1.5\times 0.5Q_{imp}\ "+"\ 1.5\times 0.5Q_{wind}$$
$$=\ 1.35G\ "+"\ 0.75Q_{imp}\ "+"\ 0.75Q_{wind}$$

Floor loads

$$1.35G\ "+"\ 1.5\times 0.7Q_{imp}\ "+"\ 1.5\times 0.5Q_{wind}$$
$$=\ 1.35G\ "+"\ 1.05Q_{imp}\ "+"\ 0.75Q_{wind}$$

Vertical loads

Design values of combined vertical loads

Roof: $q_{tot,r,d} = (3.5 \times 1.35) + (1.0 \times 0.75) = 5.48$ kN/m^2

Floor: $q_{tot,f,d} = (3.5 \times 1.35) + (6.0 \times 1.05) = 11.03$ kN/m^2

Design of bracing system in a multi-storey braced frame	Sheet 4 of 17	Rev A

Horizontal loads

Design value of total horizontal wind load (per bracing system) for load Combination 1 is:

$0.75 \times 154.0 = 115.5$ kN

Design value of wind load acting at roof level

$$= \frac{3.0 \times 0.5}{12.5} \times 115.5 = 13.86 \text{ kN}$$

Design value of wind load acting at 2nd and 3rd floor level

$$= \frac{2 \times 3.0 \times 0.5}{12.5} \times 115.5 = 27.72 \text{ kN}$$

Design value of wind load acting at 1st floor level

$$= \frac{(3.0 + 3.5) \times 0.5}{12.5} \times 115.5 = 30.03 \text{ kN}$$

Combination 1 – variable floor and roof loads as "leading" actions to 6.10b

Substituting the values of the factors on actions, (6.10b) becomes:

Roof loads

$$0.925 \times 1.35G \text{ "+" } 1.5Q_{\text{imp}} \text{ "+" } 1.5 \times 0.5Q_{\text{wind}}$$

$$= 1.25G \text{ "+" } 1.5Q_{\text{imp}} \text{ "+" } 0.75Q_{\text{wind}}$$

Floor loads

$$0.925 \times 1.35G \text{ "+" } 1.5Q_{\text{imp}} \text{ "+" } 1.5 \times 0.5Q_{\text{wind}}$$

$$= 1.25G \text{ "+" } 1.5Q_{\text{imp}} \text{ "+" } 0.75Q_{\text{wind}}$$

Vertical loads

Design values of combined vertical loads:

Roof: $q_{\text{tot,r,d}} = (3.5 \times 1.25) + (1.0 \times 1.5) = 5.88$ kN/m^2

Floor: $q_{\text{tot,f,d}} = (3.5 \times 1.25) + (6.0 \times 1.5) = 13.38$ kN/m^2

Horizontal loads

The design values of the wind loads are identical as the combination factors for wind as an "accompanying variable action" are the same in expressions (6.10a) and (6.10b).

The preceding calculations demonstrate that expression (6.10b) is more onerous, which is the common situation. For the remainder of this example, only (6.10b) will be considered when calculating design values of actions.

Combination 2 – wind load as "leading" action to 6.10b

Substituting the values of the factors on actions, 6.10b becomes:

Roof loads

$$0.925 \times 1.35G \text{ "+" } 1.5Q_{\text{wind}} \text{ "+" } 1.5 \times 0.5Q_{\text{imp}}$$

$$= 1.25G \text{ "+" } 1.5Q_{\text{wind}} \text{ "+" } 0.75Q_{\text{imp}}$$

Floor loads

$$0.925 \times 1.35G \text{ "+" } 1.5Q_{\text{wind}} \text{ "+" } 1.5 \times 0.7Q_{\text{imp}}$$

$$= 1.25G \text{ "+" } 1.5Q_{\text{wind}} \text{ "+" } 1.05Q_{\text{imp}}$$

Vertical loads

Design values of combined vertical loads:

Roof: $q_{tot,r,d} = (3.5 \times 1.25) + (1.0 \times 0.75) = 5.13$ kN/m^2

Floor: $q_{tot,f,d} = (3.5 \times 1.25) + (6.0 \times 1.05) = 10.68$ kN/m^2

Horizontal loads

Design value of total horizontal wind load (per bracing system) for Combination 2 is:

$1.5 \times 154.0 = 231.0$ kN

Design value of wind load acting at roof level

$$= \frac{3.0 \times 0.5}{12.5} \times 231.0 = 27.72 \text{ kN}$$

Design value of wind load acting at 2nd and 3rd floor level

$$= \frac{2 \times 3.0 \times 0.5}{12.5} \times 231.0 = 55.44 \text{ kN}$$

Design value of wind load acting at 1st floor level

$$= \frac{(3.0 + 3.5) \times 0.5}{12.5} \times 231.0 = 60.06 \text{ kN}$$

The design forces on an internal column are presented below in Table A.1.

Table A.1 *Design values of combined vertical forces*

	G kN/m^2	Q_{imp} kN/m^2	Load Combination 1 kN/m^2	Load Combination 2 kN/m^2
Roof	3.5	1.0	5.88	5.13
Floor	3.5	6.0	13.38	10.68

Combination 1 should be used for determining the vertical loads acting on columns not contributing to the bracing system. Columns forming the bracing system should be checked under both combinations.

A.6 Determination of design forces in columns at ULS

Columns can be classed according to their plan location:

Internal columns

Edge columns

Corner columns.

The building is designed based on the assumption of "simple construction" where only the braced frames attract and resist horizontal wind loads. The non-braced internal, edge and corner columns resist only permanent and imposed loads from the building floors. The calculations of the design loads for an internal column is shown below.

Reduction factors for multi-storey buildings

Two reduction factors are potentially available to reduce the variable vertical loads.

BS EN 1991-1-1 NA.2.6

1. BS EN 1991-1-1, 6.3.1.2 (10) allows a reduction factor α_A, which accounts for large floor areas.
2. BS EN 1991-1-1, 6.3.1.2 (11) allows a reduction factor α_n, which accounts for the number of storeys.

Both reduction factors are modified in the NA. Reductions are not available if the loading has been specifically determined.

BS EN 1991-1-1, 3.3.2 (2) specifies that if the imposed load is an accompanying action, only one of the factors, ψ or α_n may be used. Thus in Combination 2, where ψ has been applied to the imposed load as an accompanying action, α_n cannot be used. In combination 1, where ψ has not been applied to the imposed load, α_n may be used.

BS EN 1991-1 NA.2.5 gives the following expression for α_A:

$\alpha_A = 1.0 - A/1000 \geq 0.75$, where A is the area supported in m^2.

For areas above 250 m^2 the reduction factor is limited to 0.75

BS EN 1991-1 NA.2.6 gives the following expression for α_n:

$$\alpha_n = 1.1 - \frac{n}{10} \quad \text{for} \quad 1 \leq n \leq 5$$

Where n is the number of storeys with loads qualifying for reduction.

NA.2.6 specifies that reductions based on NA.2.5 may be applied if $\alpha_A < \alpha_n$ but both reductions cannot be applied simultaneously.

The appropriate load reductions are therefore:

In Combination 1, the more advantageous of either α_A or α_n may be used, but not both.

In Combination 2, only α_n may be used.

In practice, it may be simpler to ignore load reductions. It is recommended that to avoid complexity, this reduced loading is not used when considering frame imperfections and to determine equivalent horizontal forces when considering sway stability.

Column forces, Combination 1

An internal column is assumed to support a floor area of 7 m × 7 m (49 m^2). Hence the design vertical forces from the roof and each of the floors based on expression (6.10b) and Combination 1 are:

Roof:

Design value of force due to permanent load

= 3.5 kN/m^2 × 1.25 × 49.0 m^2 = 214.4 kN

Design value of force due to variable loads

= 1.0 kN/m^2× 1.50 × 49.0 m^2 = 73.5 kN

Floors:

Design value of force due to permanent load

= 3.5 × 1.25 × 49.0 = 214.4 kN

Design value of force due to variable loads

= 6.0 × 1.50 × 49.0 = 441.0 kN

Reduction factors

In combination 1, either α_A or α_n may be used, but not both.

$\alpha_A = 1.0 - A/1000 \geq 0.75$

$\alpha_A = 1.0 - (49 \times 28)/1000 \geq 0.75 = 0.75$

$\alpha_n = 1.1 - \frac{n}{10}$ and varies with the number of storeys.

At ground level, 4 storeys are supported; $\alpha_n = 1.1 - \frac{4}{10} = 0.7$

Table A.2 *Column loads based on reduced imposed loading assumptions*

	Design force due to *G* (kN)	Design force due to *Q* (kN)	Force in column due to *Q* (kN)	Reduction factor α_A	Reduction factor α_n	Minimum factor	Reduced force due to *Q* (kN)	Design force in column (kN)
Roof	214.4	73.5						
			73.5	0.75	1.0	0.75	55.1	269.5
3rd floor	214.4	441.0						
			514.5	0.75	0.9	0.75	385.9	814.7
2nd floor	214.4	441.0						
			955.5	0.75	0.8	0.75	716.6	1359.8
1st floor	214.4	441.0						
			1396.5	0.75	0.7	0.7	977.6	1835.2

Column forces, Combination 2

The design vertical forces from the roof and each of the floors based on expression 6.10b and Combination 2 are:

Roof:

Design value of force due to permanent load

$= 3.5 \text{ kN/m}^2 \times 1.25 \times 49.0 \text{ m}^2 = 214.4 \text{ kN}$

Design value of force due to variable loads

$= 1.0 \text{ kN/m}^2 \times 0.75 \times 49.0 \text{ m}^2 = 36.8 \text{ kN}$

Floors:

Design value of force due to permanent load

$= 3.5 \times 1.25 \times 49.0 = 214.4 \text{ kN}$

Design value of force due to variable loads

$= 6.0 \times 1.05 \times 49.0 = 308.7 \text{ kN}$

Reduction factors

In Combination 2, only α_n may be used, since the variable actions have been factored by ψ

$\alpha_n = 1.1 - \frac{n}{10}$ and varies with the number of storeys.

Table A.3 *Column loads based on reduced imposed loading assumptions*

	Design force due to *G* (kN)	Design force due to *Q* (kN)	Force in column due to *Q* (kN)	Reduction factor α_n	Reduced force due to *Q* (kN)	Design force in column (kN)
Roof	214.4	36.8				
			36.8	1.0	33.1	247.5
3rd floor	214.4	308.7				
			345.5	0.9	311.0	739.6
2nd floor	214.4	308.7				
			654.2	0.8	523.4	1166.6
1st floor	214.4	308.7				
			962.9	0.7	674.0	1531.6

As can be seen from Table A.2 and Table A.3, the axial forces from combination 1 are more onerous, and should be used for design. From Table A.2 the column between ground level and first floor level must resist an axial compressive force of 1835.2 kN.

Chosen column and beam member sizes

For the above floor loads and column design forces, the following section sizes provide adequate resistance.

Refer to tables in P363

Roof beams 305 × 127 × 37 UB

Floor beams 406 × 178 × 60 UB

Ground to 2nd floor columns 203 × 203 × 60 UC

2nd floor to roof columns 203 × 203 × 46 UC

Assumed bracing 168.3 × 6.3 CHS

S275 steel is use throughout for UBs and UCs. S355 steel is used for hollow sections.

The same column sizes are assumed in the bracing system considered below and shown in Figure A.2 and Figure A.3

A.7 Sway stiffness

The sway stiffness of the structure is assessed by performing an elastic analysis on one of the braced bays, under the action of applied horizontal forces (wind loads) combined with the equivalent horizontal forces, according to the rules in BS EN 1993-1-1 clause 5.2.2 (3)(b), (4) and (6).

BS EN 1993-1-1 5.3.2(3)

The equivalent horizontal forces (EHF) are given by clause 5.3.2 (7), although only sway imperfections are considered (member imperfections are taken into account in the rules for verifying member resistances).

Global initial sway imperfections ϕ are given by 5.3.2(3) as:

5.3.2 (3a) Eqn (5.5)

$$\phi = \phi_0 \, \alpha_h \, \alpha_m$$

where:

ϕ_0 is 1/200

α_h is the reduction factor for height h applicable to columns

α_m is the reduction factor for the number of columns in a row

ANALYSIS AND DESIGN CALCULATIONS
FROM EMBODIED OPTIMISATION TOOL

BUILDING GEOMETRY DATA

Number of floors (along Z axis) = 4
Floor to floor height (m) = 3.5
Number of bays along along X axis = 7
Grid distance along X axis (m) = 7
Number of bays along along Y axis= 4
Grid distance along Y axis (m) = 7
Floor slab span (m) = 7

ACTIONS

1A. Vertical loads on roof:

Permanent G (kN/m2) = 3.5
Snow Q1 (kN/m2) = 1

1B. Vertical loads on floor:

Permanent G (kN/m2) = 3.5
Imposed Q1 (kN/m2) = 6

2. Horizontal loads:

Wind load Q2 (kN/m2) = 0.88

LOAD FACTORS

1. Partial Factors (Gamma):

For permanent loads Gamma_G = 1.35
For variable loads Gamma_Q = 1.5

2. Combination Factors (Psi_0):

For imposed loads Psi_0_1 = 0.7
For snow loads Psi_0_1 = 0.5
For wind loads Psi_0_2 = 0.5

3. Reduction Factors:

For permanent loads Xi = 0.925
For imposed loads for design of columns/walls, accounts for number of storeys
Alpha_n_roof = 1
Alpha_n_floor#3 = 0.9
Alpha_n_floor#2 = 0.8
Alpha_n_floor#1 = 0.7

Begin Analysis...

ULTIMATE LIMIT STATE LOAD COMBINATION EQUATIONS

1. (Gamma_G * G) + (Gamma_Q * Psi_0_1 * Q1) + (Gamma_Q * Psi_0_2 * Q2)
2. (Gamma_G * Xi * G) + (Gamma_Q * Q1) + (Gamma_Q * Psi_0_2 * Q2)
3. (Gamma_G * Xi * G) + (Gamma_Q * Psi_0_1 * Q1) + (Gamma_Q * Q2)
ULS design forces is the least favorable (maximum) value of the 3 combinations

From the 3 combinations, only terms containing:
- G and Q1 are considered in determining vertical design loads
- Q2 are considered in determining horizontal design loads

Beams
Span of secondary beams = Grid distance along Y axis
Span of primary beams = Grid distance along X axis

Columns
Area supported by middle columns = 2 * Area supported by edge columns
For columns, Q1 becomes Q1_reduced = Q1 * Alpha_n

Wind bracing
Each facade (2 along length, 2 along width of building) has wind bracings for stability

Ultimate Limit State Design Loads
Design load for primary (middle) beams supporting:
Roof = 41.094 kN/m
Floor#3 = 93.594 kN/m
Floor#2 = 93.594 kN/m
Floor#1 = 93.594 kN/m
Design load for (middle) columns supporting:
Roof = 292.232 kN
Floor#3 = 899.422 kN
Floor#2 = 1414.933 kN
Floor#1 = 1842.243 kN
Design load for diagonals in wind bracing:
Bracing along length of building = 289.258 kN
Bracing along width of building = 506.201 kN

SERVICEABILITY LIMIT STATE LOAD COMBINATION EQUATIONS
1. G + Q1 + (Psi_0_2 * Q2)
2. G + (Psi_0_1 * Q1) + Q2
SLS design forces is the least favorable (maximum) value of the 2 combinations
From the 2 combinations, only terms containing G and Q1 are considered in determining vertical design loads

Beams
Span of secondary beams = Grid distance along Y axis
Span of primary beams = Grid distance along X axis

Serviceability Limit State Design Loads
Design load for primary middle beams supporting:
Roof = 31.5 kN/m
Floor#3 = 66.5 kN/m
Floor#2 = 66.5 kN/m

Floor#1 = 66.5 kN/m

End of Analysis!

Begin Design...

Primary beams

Roof 457x152x52 S275 (52.3 kg/m)
Floor #3 533x210x92 S275 (92.1 kg/m)
Floor #2 533x210x92 S275 (92.1 kg/m)
Floor #1 533x210x92 S275 (92.1 kg/m)

Columns

Roof 152x152x23 S275 (23 kg/m)
Floor #3 356x171x45 S275 (45 kg/m)
Floor #2 203x203x60 S275 (60 kg/m)
Floor #1 203x203x71 S275 (71 kg/m)

Wind bracing

Diagonals along length of building CHSC88.9x4.0 S275 (8.4 kg/m)
Diagonals along width of building CHSH168.3x3.6 S275 (14.6 kg/m)

Calculation in Detail:

Design of Primary beams

> ULS design Load = 41.09 kN/m
> Beam span = 7 m
> Design shear force = Design load * Beam span / 2 = 143.83 kN
> Design moment = Design load * Beam span ^ 2 / 8 = 251.7 kNm
> Yield strength = 275 N/mm2
> Partial factor = 1
> Plastic modulus required = design moment * partial factor / yield strength = 915.28 cm3
> Maximum allowed deflection = Beam span / 250 = 28 mm
> SLS design Load = 31.5 kN/m
> Choosing section profile 457x152x52
> Cross-section class of section profile = 1
> Height of section profile = 449.8 mm
> Web-thickness of section profile = 7.6 mm
> Moment of inertia Y of section profile = 213690000 mm4
> Plastic modulus Y of section profile = 1092.5 cm3 >= Plastic modulus required, therefore, OK
> Shear resistance of section profile = 1.04 * Height * Web-thickness * (Yield strength / 1.732) * Partial factor = 564.47 kN >= Design shear force, therefore, OK
> Deflection in beam = 5 / 384 * SLS design load * Beam span ^ 4 / Modulus of elasticity * Moment of inertia Y = 21.95 mm <= Maximum allowed deflection, therefore, OK

Design of Primary beams

> ULS design Load = 93.59 kN/m
> Beam span = 7 m
> Design shear force = Design load * Beam span / 2 = 327.58 kN
> Design moment = Design load * Beam span ^ 2 / 8 = 573.27 kNm
> Yield strength = 275 N/mm2
> Partial factor = 1
> Plastic modulus required = design moment * partial factor / yield strength = 2084.6 cm3
> Maximum allowed deflection = Beam span / 250 = 28 mm
> SLS design Load = 66.5 kN/m
> Choosing section profile 533x210x92

> Cross-section class of section profile = 1
> Height of section profile = 533.1 mm
> Web-thickness of section profile = 10.1 mm
> Moment of inertia Y of section profile = 552270000 mm4
> Plastic modulus Y of section profile = 2382.8 cm3 >= Plastic modulus required, therefore, OK
> Shear resistance of section profile = 1.04 * Height * Web-thickness * (Yield strength / 1.732) * Partial factor = 889.07 kN >= Design shear force, therefore, OK
> Deflection in beam = 5 / 384 * SLS design load * Beam span ^ 4 / Modulus of elasticity * Moment of inertia Y = 17.93 mm <= Maximum allowed deflection, therefore, OK

Design of Primary beams

> ULS design Load = 93.59 kN/m
> Beam span = 7 m
> Design shear force = Design load * Beam span / 2 = 327.58 kN
> Design moment = Design load * Beam span ^ 2 / 8 = 573.27 kNm
> Yield strength = 275 N/mm2
> Partial factor = 1
> Plastic modulus required = design moment * partial factor / yield strength = 2084.6 cm3
> Maximum allowed deflection = Beam span / 250 = 28 mm
> SLS design Load = 66.5 kN/m
> Choosing section profile 533x210x92
> Cross-section class of section profile = 1
> Height of section profile = 533.1 mm
> Web-thickness of section profile = 10.1 mm
> Moment of inertia Y of section profile = 552270000 mm4
> Plastic modulus Y of section profile = 2382.8 cm3 >= Plastic modulus required, therefore, OK
> Shear resistance of section profile = 1.04 * Height * Web-thickness * (Yield strength / 1.732) * Partial factor = 889.07 kN >= Design shear force, therefore, OK
> Deflection in beam = 5 / 384 * SLS design load * Beam span ^ 4 / Modulus of elasticity * Moment of inertia Y = 17.93 mm <= Maximum allowed deflection, therefore, OK

Design of Primary beams

> ULS design Load = 93.59 kN/m
> Beam span = 7 m
> Design shear force = Design load * Beam span / 2 = 327.58 kN
> Design moment = Design load * Beam span ^ 2 / 8 = 573.27 kNm
> Yield strength = 275 N/mm2
> Partial factor = 1
> Plastic modulus required = design moment * partial factor / yield strength = 2084.6 cm3
> Maximum allowed deflection = Beam span / 250 = 28 mm
> SLS design Load = 66.5 kN/m
> Choosing section profile 533x210x92
> Cross-section class of section profile = 1
> Height of section profile = 533.1 mm
> Web-thickness of section profile = 10.1 mm
> Moment of inertia Y of section profile = 552270000 mm4
> Plastic modulus Y of section profile = 2382.8 cm3 >= Plastic modulus required, therefore, OK
> Shear resistance of section profile = 1.04 * Height * Web-thickness * (Yield strength / 1.732) * Partial factor = 889.07 kN >= Design shear force, therefore, OK
> Deflection in beam = 5 / 384 * SLS design load * Beam span ^ 4 / Modulus of elasticity * Moment of inertia Y = 17.93 mm <= Maximum allowed deflection, therefore, OK

Design of Columns

> ULS Design Load = 292.23 kN
> Column span = 3.5 m
> Yield strength = 275 N/mm2
> Buckling length = 3.5 m
> Partial factor = 1
> Choosing section profile 152x152x23
> Cross-section class of section profile = 3
> Radius of gyration Z = 37 mm

> Modulus of elasticity = 210000 N/mm2
> Slenderness ratio = Modulus of elasticity / Yield strength = 86.81
> Non-dimensional slenderness = Buckling length / Radius of gyration / Slenderness ratio = 1.09
> Reduction factor buckling = 0.54
> Design buckling resistance = Reduction factor buckling * Area * Yield strength / Partial factor = 434.79 kN <= ULS Design load, therefore, OK

Design of Columns

> ULS Design Load = 899.42 kN
> Column span = 3.5 m
> Yield strength = 275 N/mm2
> Buckling length = 3.5 m
> Partial factor = 1
> Choosing section profile 356x171x45
> Cross-section class of section profile = 2
> Radius of gyration Z = 37.6 mm
> Modulus of elasticity = 210000 N/mm2
> Slenderness ratio = Modulus of elasticity / Yield strength = 86.81
> Non-dimensional slenderness = Buckling length / Radius of gyration / Slenderness ratio = 1.07
> Reduction factor buckling = 0.62
> Design buckling resistance = Reduction factor buckling * Area * Yield strength / Partial factor = 969.32 kN <= ULS Design load, therefore, OK

Design of Columns

> ULS Design Load = 1414.93 kN
> Column span = 3.5 m
> Yield strength = 275 N/mm2
> Buckling length = 3.5 m
> Partial factor = 1
> Choosing section profile 203x203x60
> Cross-section class of section profile = 1
> Radius of gyration Z = 52 mm
> Modulus of elasticity = 210000 N/mm2
> Slenderness ratio = Modulus of elasticity / Yield strength = 86.81
> Non-dimensional slenderness = Buckling length / Radius of gyration / Slenderness ratio = 0.78
> Reduction factor buckling = 0.74
> Design buckling resistance = Reduction factor buckling * Area * Yield strength / Partial factor = 1553.82 kN <= ULS Design load, therefore, OK

Design of Columns

> ULS Design Load = 1842.24 kN
> Column span = 3.5 m
> Yield strength = 275 N/mm2
> Buckling length = 3.5 m
> Partial factor = 1
> Choosing section profile 203x203x71
> Cross-section class of section profile = 1
> Radius of gyration Z = 53 mm
> Modulus of elasticity = 210000 N/mm2
> Slenderness ratio = Modulus of elasticity / Yield strength = 86.81
> Non-dimensional slenderness = Buckling length / Radius of gyration / Slenderness ratio = 0.76
> Reduction factor buckling = 0.75
> Design buckling resistance = Reduction factor buckling * Area * Yield strength / Partial factor = 1860.46 kN <= ULS Design load, therefore, OK

Design of diagonal in wind bracing along length of building

> ULS Design Load = 289.26 kN
> Diagonal span = 7.83 m
> Yield strength = 275 N/mm2
> Partial factor = 1
> Choosing section profile CHSC88.9x4.0
> Design axial resistance = Area * Yield strength / Partial factor = 294.25 kN <= ULS Design load, therefore, OK

Design of diagonal in wind bracing along width of building

> ULS Design Load = 506.2 kN
> Diagonal span = 7.83 m
> Yield strength = 275 N/mm2
> Partial factor = 1
> Choosing section profile CHSH168.3x3.6
> Design axial resistance = Area * Yield strength / Partial factor = 511.5 kN <= ULS Design load, therefore, OK

End of Design!