
INTEGRATING BUILDING INFORMATION MODELLING WITH LIFE-CYCLE ASSESSMENT AND LIFE-CYCLE COSTING

An exploration into the potentials of an integrated LCA-LCC application with BIM for the determination and optimisation of environmental impact and cost analysis of construction materials during early design stages

Student	C.D. (Casper) van Oeveren
ID Number	1036233
Document	Thesis

This page is intentionally left blank

COLOPHON

TITLE

Integrating Building Information Modelling with Life-Cycle Assessment and Life-Cycle Costing

An exploration into the potentials of an integrated LCA-LCC application with BIM for the determination and optimisation of environmental impact and cost analysis of construction materials during early design stages

DOCUMENT Master thesis
DATE 04-03-2020
VERSION V1.0

AUTHOR C.D. (Casper) van Oeveren
ID NUMBER 1036233
ADDRESS Doctor Struyckenstraat 60
4812 BG Breda
PHONE NUMBER +31 (0)6 27 25 22 47
TUE-EMAIL c.d.v.oeveren@student.tue.nl
Email caspervanoeveren@gmail.com
LINKED-IN linkedin.com/in/casper-van-oeveren/

INSTITUTE University of Technology Eindhoven
FACULTY Department of the Built Environment
MASTER PROGRAM Construction Management and Engineering
ADDRESS Den Dolech 2
5612 AZ Eindhoven

GRADUATION COMPANY Van de Ven (VolkerWessels)
ADDRESS Haya van Somerenstraat 4-01
5032 MK Tilburg

CHAIRMAN Prof.dr.ir. B. de Vries
GRADUATION SUPERVISOR 1ST dr. Q. Han
GRADUATION SUPERVISOR 2ND Ir. L. Wijnholts
COMPANY SUPERVISORS A. van Hulten
M. van den Berg

*“We must accept finite disappoint,
but never lose infinite hope.”
– Martin Luther King, Jr*

*“Simplicity is the ultimate form of sophistication”
– Leonardo Da Vinci*

PREFACE

After a graduation period of six months, I can finally present you my graduation thesis on the exploration into the potentials of integrating Life-Cycle Assessment (LCA) and Life-Cycle Costing (LCC) with Building Information Modelling (BIM) for the determination and optimisation of environmental impact and cost analysis of construction materials during the early design stages. This graduation thesis has been written in fulfilment of the master Construction Management and Engineering at the Eindhoven University of Technology (TU/e).

During my entire period at the TU/e, I have known my ups and downs. The past six months were not different from the rest. Inspired by the many classes during my master's program, I wanted to push me to the limits one last time. Therefore, I decided to do my research on three promising subjects within the AEC-industry, being BIM, LCA and LCC. In addition, I challenged myself to learn how to program using Python. Based on the minor expertise in the fields of LCA and LCC, and even no experience in programming, I tried to get familiar with these subjects during my graduation. The gained knowledge of LCA, LCC and BIM, as well as Python, will hopefully help me to achieve my ambition, which is providing people useful information on how to become more sustainably aware of their decisions.

I would like to give my gratitude to some people that supported me during the graduation process without whom I would not be able to finish my thesis. First of all, I would like to thank my university supervisors Qi Han and Luuk Wijnholts, for their guidance and useful insights during the entire graduation process. With regards to my graduation company, I would like to give my gratitude to Alex van Hulten and Martijn van den Berg, who guided me and provided me with a lot of practical knowledge. I also would like to thank my family and friends for the enormous support I have received from them. In particular, I would like to thank Lieke for all the mental support she provided. Without you, I could not have finished or even start my thesis.

I am sure that I forgot some people. For all my fellow students, PhD-students, and co-workers at Van de Ven Bouw en Ontwikkeling who supported and guided me through the graduation process, thank you.

To all who have taken the time to read this preface, thank you. I wish you a lot of reading pleasure.

Casper van Oeveren

This page is intentionally left blank

CONTENTS

SUMMARY	III
SAMENVATTING	V
ABSTRACT.....	VII
GLOSSARY	VIII
LIST OF FIGURES	X
LIST OF TABLES.....	X
LIST OF EQUATIONS.....	XI
1 INTRODUCTION	1
1.1 RESEARCH CONTEXT	2
1.2 PROBLEM STATEMENT	5
1.3 RESEARCH OBJECTIVES	5
1.4 RESEARCH QUESTION.....	6
1.5 RESEARCH DESIGN	7
1.6 READING GUIDE.....	8
2 LINKING THE DOTS: LCA, LCC AND BIM.....	9
2.1 LIFE-CYCLE ASSESSMENT AND LIFE-CYCLE COSTING.....	10
2.2 THE ROLE OF LIFE-CYCLE ASSESSMENT AND LIFE-CYCLE COSTING IN THE AEC-INDUSTRY	13
2.3 BUILDING INFORMATION MODELLING	23
2.4 THE INTEGRATION OF LCA, LCC AND BIM	26
2.5 FEEDBACK APPROACHES	33
2.6 CHALLENGES OF INTEGRATION.....	34
3 A BIM-BASED LIFE-CYCLE ENVIRONMENTAL AND ECONOMIC IMPACT ASSESSMENT TOOL	39
3.1 REQUIREMENTS FOR INTEGRATING LIFE-CYCLE ENVIRONMENTAL AND COST ASSESSMENT WITH BIM	40
3.2 THE BASE OF THE APPLICATION	42
3.3 THE SYSTEM'S DESIGN.....	44
3.4 THE INPUT.....	47
3.5 THE PROCESS	51
3.6 THE OUTPUT.....	54
3.7 THE FEEDBACK	57
3.8 FIELD OF IMPROVEMENTS.....	58
4 CASE STUDY - APARTMENT BUILDING	60
4.1 GOAL AND SCOPE DEFINITION.....	61
4.2 INVENTORY ANALYSIS	61
4.3 IMPACT ASSESSMENT.....	62
4.4 INTERPRETATION STAGE	63
5 CONCLUSION	67
5.1 THE RESEARCH QUESTION	68
5.2 DISCUSSION.....	71
5.3 RECOMMENDATION	72

6	REFERENCES.....	75
	APPENDICES.....	85
	APPENDIX I - MPG-METHOD	85
	APPENDIX II - LIST OF INTERVIEWEES	90
	APPENDIX III - PYTHON SCRIPTS	91
	APPENDIX IV - ENVIRONMENTAL PRODUCT DECLARATIONS	168
	APPENDIX V - THE RESULTS OF THE DIFFERENT VARIATIONS OF THE CASE STUDY	169

SUMMARY

Over the past 50 years, the urge to become more sustainably aware is rising. It is acknowledged that the earth's raw resources are diminishing rapidly. It can be stated that AEC-industry is a major contributor to the depletion of raw resources and puts extreme pressure natural environment. The pressure on the environment by the AEC-industry should be reduced. A way of approaching this issue is by using Life-cycle Thinking. Life-cycle Thinking or LCT is gaining a prominent role in decision-making, generally from an environmental perspective. With regards to the AEC-industry, it is crucial that when using the LCT-approach, features such as emissions control, use of materials, depletion of resources and the reuse and recycling of resources or materials should be considered. Within the LCT, this environmental approach is called Life-Cycle Assessment (LCA). In the AEC-industry, LCA currently acts as a standardised method, which is used for the assessment of environmental impacts of buildings throughout their entire life-cycle. However, a holdback for implementing LCA in the AEC-industry is the reluctance of implementing sustainability measures by the AEC-industry. One of the main reasons for not implementing LCA by the AEC-industry is that the industry is conservative and does not feel a high need for change. Besides, by looking at just the construction industry, lack of knowledge in terms of the added value of sustainability is considered to be a significant issue. Therefore, it is essential that aspects of Life-cycle Thinking other than the environmental LCA are taken into account as well, such as social effects and economic effects. Life-Cycle Costing (LCC) is an approach of LCT that embodies the financial costs of materials and products that makes investment costs, maintenance costs and demolition costs over the entire life-cycle of a building. By combining LCA with LCC, the AEC-industry gets a better understanding of what both the environmental as well as the economic benefits are. Therefore, analysing and optimising the embodied life-cycle-related impacts of building materials with regards to LCA and LCC has become an essential field of research for the implementation of LCT in, for example, the AEC-industry.

However, the application of LCA and LCC in the AEC industry is limited because executing LCA and LCC studies is very time-consuming. Besides, this implementation of LCA and LCC studies requires a high level of knowledge; in general, only experts can carry out this type of studies. That is why LCA and LCC studies are usually carried out after the completion of the building since, in that case, all information is final and complete. On the one hand, this has the advantage that a complete description can be made about the true nature of the environmental and economic impacts of this building. However, this means that the application of the LCA and LCC is only descriptive rather than supportive in terms of providing feedback during the design process to enhance and optimise the design. As a result, the way of working results in an inefficient way of modifying a design, which means that achieving specific environmental performance criteria can be time-consuming.

A way of overcoming the descriptive nature of LCA and LCC with regards to the AEC industry is the utilisation of Building Information Modelling (BIM). By utilising BIM, specifically IFC-models, in combination with LCA and LCC, the execution of LCA and LCC studies in the early design stages can be simplified. This integration of features is especially important in early design stages, since during these stages, in general, most changes in design occur through the many study alternatives. Having an integration of IFC-models, LCA and LCC ensure that changes are implemented, checked for possible environmental and economic impacts, and, if

necessary, other adjustments made to ensure that all aspects meet the legal requirements and wishes from the different project participants. In other words, this integration ensures that through an iterative process, the building design can be optimised in the early design stages concerning the environmental and economic impacts of a building design. It must be stated that BIM can be seen as a tool, whereby the facilitation of bill of quantities can ensure that timely insight is provided about potential environmental and economic bottlenecks along the design process. Many types of research, therefore, suggest that BIM is a way of working to ensure that the environmental performance of buildings and the economic impact of these buildings can be optimised and improved during the early design stages.

At present times, there are few applications on the market that help non-experts to test designs in the field of LCA and LCC. These applications are often made to show the environmental impacts of designs only. Only a handful of applications also show the economic impacts of the designs. However, these programs are generally not user-friendly and do not have any feedback-related features for optimising designs. To optimise the application of LCA and LCC in the early design stages, specifically in the AEC-industry, an application has been developed which enables users to get a better understanding of the embodied effects of their decisions in design with regards to LCA and LCC and provides feedback how to improve the use of materials in environmental, economic as well as a combination of the two aspects. The proposed application is called LEICAS, which is an abbreviation for Life-Cycle Environmental Impact and Costing Assessment.

In the beginning stages of a project, IFC-models are mainly not present. Therefore, the application does not require an IFC-model to be used; a simple Excel-sheet can be filled in containing certain aspects such as categorisation of the materials, square metres, and element widths. In the case an IFC-model is present, the LEICAS-application automatically extracts the bill of quantities (BoQ) of the building elements and categorises them by their identification code. This categorisation of materials is based on the Dutch NL/Sfb coding system. Based on this categorisation, the BoQ is linked to an external LCA and LCC database, which contains the environmental and economic data of materials that can be linked to the same categorisation schema of the IFC-model. Overall, the LEICAS application is particularly suitable for use in the early design stages, in which a design usually is insufficiently elaborated on, and the potential to reduce the embodied impacts is highest.

By executing a case study, it can be stated that it is possible to assess the different impacts of a design at an early stage in order to improve and optimise the building design. The application can import IFC-models, combine them with an external LCA and LCC database and ensure that the results are displayed in a user-friendly interface. Through this interface, users can gain insight into the meaning of the results regarding the environmental and economic impacts of their designs and how they can optimise and improve certain building elements in their design. The LEICAS application proves that an initial step towards the implementation of LCA and LCC in the AEC industry is possible.

SAMENVATTING

In de afgelopen 50 jaar neemt de drang om duurzamer te worden toe. Onderzoeken hebben aangetoond dat de ruwe grondstoffen van de aarde snel afnemen. Er kan worden gesteld dat de AEC-industrie een belangrijke bijdrage levert aan de uitputting van grondstoffen en een extreme druk legt op het milieu. Een manier om dit probleem aan te pakken, is Life-Cycle Thinking (LCT). LCT krijgt een prominente rol in de besluitvorming, meestal vanuit milieutechnisch oogpunt. Binnen de AEC-industrie is het van cruciaal belang dat bij het gebruik van de LCT-aanpak rekening wordt gehouden met kenmerken zoals emissiebeheersing, materiaalgebruik, uitputting van grondstoffen en de recycling en hergebruik van materialen. Binnen de LCT wordt deze milieuaanpak ook wel Life-Cycle Assessment (LCA) genoemd. In de AEC-industrie fungeert LCA momenteel als een gestandaardiseerde methode, die wordt gebruikt voor de beoordeling van de milieueffecten van gebouwen gedurende hun volledige levenscyclus. Echter, op het gebied van LCA-implementatie is de AEC-industrie echter terughoudendheid. Een van de belangrijkste redenen vanuit de AEC-industrie om LCA niet te implementeren, is dat de industrie conservatief is en geen grote behoefte aan verandering heeft. Door alleen al naar de bouwsector te kijken, wordt bovendien een gebrek aan kennis over de toegevoegde waarde van duurzaamheid als een belangrijk probleem beschouwd. Het is daarom van essentieel belang dat ook andere aspecten van Life-cycle Thinking worden beschouwd, zoals sociale en economische effecten. Life-Cycle Costing (LCC) is een benadering van LCT die de financiële kosten van materialen en producten belichaamt die investeringskosten, onderhoudskosten en sloopkosten over de gehele levenscyclus van een gebouw bij elkaar brengt. Door LCA met LCC te combineren, krijgt de AEC-industrie een beter inzicht in wat zowel de milieutechnische als de economische voordelen zijn. Daarom is het analyseren en optimaliseren van de belichaamde levenscyclus gerelateerde effecten van bouwmaterialen met betrekking tot LCA en LCC een essentieel onderzoeksgebied geworden voor de implementatie van LCT in bijvoorbeeld de AEC-industrie.

De toepassing van LCA en LCC in de AEC-industrie is echter beperkt omdat het uitvoeren van LCA- en LCC-onderzoeken als zeer tijdrovend wordt beschouwd. Bovendien vereist deze implementatie van LCA- en LCC-onderzoeken een hoog kennisniveau; over het algemeen kunnen enkel deskundigen dit soort onderzoeken uitvoeren. Dat is de reden waarom deze onderzoeken meestal worden uitgevoerd na de voltooiing van het gebouw, omdat vanaf dat moment alle informatie definitief en volledig bekend zijn. Enerzijds heeft dit het voordeel dat een complete beschrijving gegeven kan worden over de ware aard van de economische en milieutechnische gevolgen van een gebouw. Dit betekent echter dat de toepassing van LCA en LCC alleen beschrijvend is in plaats van ondersteunend, wat tot direct gevolg heeft dat de manier van werken inefficiënt is om een ontwerp te wijzigen tijdens het ontwerpproces.

Een manier om het beschrijvende karakter van LCA en LCC onderzoeken te overwinnen in de AEC-industrie, is door het gebruik van Building Information Modelling (BIM). Door gebruik te maken van BIM, in het bijzonder IFC-modellen, in combinatie met LCA en LCC, kan de uitvoering van LCA- en LCC-onderzoeken in de vroege ontwerpfasen worden mogelijk gemaakt en vereenvoudigd. Deze integratie van functies is vooral belangrijk in vroege ontwerpfasen doordat tijdens deze fasen in het algemeen de meeste ontwerpwijzigingen plaatsvinden door de vele ontwerpalternatieven. Door middel van het gebruiken van IFC-modellen kunnen LCA en LCC ervoor zorgen dat veranderingen worden uitgevoerd in een ontwerp, deze wijzigingen

worden gecontroleerd op mogelijke milieu- en economische effecten, en, indien nodig, weer de nodige aanpassingen worden gemaakt om ervoor te zorgen dat alle aspecten voldoen aan de wettelijke vereisten en wensen van verschillende projectdeelnemers. Met andere woorden, deze integratie zorgt ervoor dat het gebouwontwerp door een iteratief proces kan worden geoptimaliseerd in de vroege ontwerpfasen met betrekking tot de milieu- en economische impact van een gebouwontwerp. Opgemerkt moet worden dat BIM moet worden gezien als een hulpmiddel, waardoor het faciliteren van een uittrekstaat kan zorgen voor tijdig inzicht in mogelijke ecologische en economische knelpunten in het ontwerpproces. Veel soorten onderzoek suggereren daarom dat BIM een manier is om ervoor te zorgen dat de milieuprestaties van gebouwen en de economische impact van deze gebouwen tijdens de vroege ontwerpfasen kunnen worden geoptimaliseerd en verbeterd.

Op dit moment zijn er veelal applicaties op de markt die alleen kunnen worden gebruikt door experts. Deze applicaties zijn ontwikkeld vanuit het oogpunt om enkel de milieueffecten van ontwerpen in kaart te brengen. Slechts een handvol toepassingen laat ook de economische impact van de ontwerpen zien. Deze applicaties zijn echter over het algemeen niet gebruiksvriendelijk en hebben geen feedback-gerelateerde functies voor het optimaliseren van ontwerpen. Om de toepassing van LCA en LCC in de vroege ontwerpfasen te optimaliseren, met name in de AEC-industrie, is een applicatie ontwikkeld waarmee gebruikers een beter inzicht krijgen in de effecten van hun beslissingen in het ontwerp met betrekking tot LCA en LCC. Daarnaast geeft deze applicatie feedback over hoe het gebruik van de huidige materialen in een ontwerp, lettende op zowel ecologisch gebied, economisch gebied en een combinatie van beide gebieden kan worden verbeterd. De voorgestelde toepassing heet LEICAS, wat een Engelse afkorting is voor de beoordeling van milieueffecten en kostenberekening gedurende de levenscyclus.

In de beginfase van een project zijn IFC-modellen meestal niet aanwezig. Daarom heeft de applicatie geen IFC-model nodig om te functioneren; een eenvoudig Excel-blad kan worden ingevuld dat bepaalde aspecten bevat, zoals categorisatie van de materialen, vierkante meters en elementdiktes. In het geval dat een IFC-model aanwezig is, haalt de LEICAS-applicatie automatisch de uittrekstaten van de bouwelementen op en categoriseert deze op basis van hun identificatiecode. Deze indeling van materialen is gebaseerd op het Nederlandse NL/Sfb coderingssysteem. Op basis van deze categorisatie worden de uittrekstaten gekoppeld aan een externe LCA- en LCC-database, die de milieu- en economische gegevens bevat van materialen. Over het algemeen is de LEICAS-applicatie bijzonder geschikt voor gebruik in de vroege ontwerpfasen, waarin een ontwerp meestal onvoldoende is uitgewerkt en het potentieel om de ecologische en economische effecten te verminderen het grootst is.

Door het uitvoeren van een casus kan worden gesteld dat het mogelijk is om de verschillende effecten van een ontwerp in een vroeg stadium te kunnen beoordelen om het gebouwontwerp te verbeteren en te optimaliseren. De applicatie importeert IFC-modellen, combineert de data met een externe LCA- en LCC-database en zorgt ervoor dat de resultaten worden weergegeven in een gebruiksvriendelijke interface. Via deze interface kunnen gebruikers inzicht krijgen in de betekenis van de resultaten met betrekking tot de milieu- en economische impact van hun ontwerpen en hoe ze bepaalde bouwelementen in hun ontwerp kunnen optimaliseren en verbeteren. De LEICAS-applicatie bewijst dat een eerste stap naar de implementatie van LCA en LCC in de AEC-industrie mogelijk is.

ABSTRACT

PURPOSE

Life-cycle Thinking (LCT) is gaining a prominent role in building design assessment for improving the design, generally from environmental and economic perspectives. Also, building information modelling (BIM) has become a powerful tool to make these analyses fast, easy and accurate. However, currently, there is a lack of building design assessment applications, which can be applied for the assessment of environmental and economic impact within different stages of design simultaneously. Furthermore, project participants generally lack in understanding the results of the assessment applications, which can lead to misinterpretation of the results and potential deficient decision-making. Therefore, the primary objective of this study is to show that a BIM-based LCA and LCC application enables designers to optimise their designs based on well-considered environmental and economic decisions.

METHODOLOGY

The research aims to establish a BIM-based building assessment application that integrates the methodologies of Life-Cycle Assessment (LCA) and Life-Cycle Costing (LCC) to calculate the environmental and economic impact of building materials in design in the early design stages. To enhance the application of LCA and LCC in the early design stages, a user-friendly application was developed that enables designers to comprehend the embodied impacts of their designs quickly. It can be used in both the conceptual design stage and the developed design stage.

FINDINGS

The case study shows that the developed application can provide results for both LCA and LCC assessment related to the environmental and economic impact of a building at the same time in the early design stage. Besides, IFC-extracts could be interpreted, and results were presented through an intuitive user interface. It can also provide component recommendations in the developed design stage. This application can be used by designers, project managers and suppliers addressing their particular focus and preferences.

ORIGINALITY

This thesis has tried to illustrate the importance of integrating BIM within LCA and LCC approaches at the early stages of a design in order to assess the embodied environmental and economic impact of building elements. The proposed application differs from other BIM, LCA and LCC integrated solutions since it generates results based on the environmental and economic impact of an IFC-model. Subsequently and most important, the application provides the users feedback on how to optimise their design by providing material selections based on environmental and economic impacts.

Building Information Modelling (BIM), Life-Cycle Assessment (LCA), Life-Cycle Costing (LCC), environmental impacts, economic impacts

GLOSSARY

ADP	Abiotic depletion potential
ADPc	Abiotic depletion potential of composites and minerals
ADPf	Abiotic depletion potential of fossil fuels
AETP	Aquatic ecotoxicity potential
AHP	Analytic Hierarchy Process
AP	Acidification potential
BCF	BIM Collaboration Format
BIM	Building Information Modelling
BoQ	Bill of quantities
CFC	Chlorofluorocarbon
CH₃Br	Methyl bromide
CH₄	Methane
CO₂	Carbon dioxide
EP	Eutrophication potential
EPD	Environmental Product Declaration
FAETP	Freshwater aquatic ecotoxicity potential
GUI	Graphical User Interface
GWP	Global warming potential
HCL	Hydrochloric acid
HF	Hydrofluoric acid
HTP	Human toxicity potential
IDM	Information Delivery Manual
IFC	Industry Foundation Classes
IPO	Input, Processing and Output
ISO	International Organization for Standardization
Kg	Kilogram
LCA	Life-cycle Analysis
LCC	Life-Cycle Costing
LCSA	Life-cycle Sustainability Assessment
LCT	Life-cycle Thinking
LEICAS	Life-Cycle Environmental and Economic Assessment Application
LOD	Level of development
m¹	Running meter
m²	Square meter
m³	Cubic meter
MAETP	Marine aquatic ecotoxicity potential
MADM	Multi-attribute decision-making
MCDM	Multi-criteria decision-making
MOLP	Multiple-objective linear programming
MODM	Multi-objective decision-making
MPG	Milieuprestatie Gebouwen
MVP	Minimum Viable Product
NH₄	Ammonia
NIBE	Nederlands Instituut voor Bouwbiologie en Ecologie
NL/SfB	Nederland/Samarbetskommittén för Byggnadsfråga

NMD	Nationale Milieudatabase
NMHC	Non-methane hydrocarbon
NO₂	Nitrogen dioxide
NO_x	Nitrogen oxides
ODP	Stratospheric ozone depletion potential
PO₄	Phosphate
POCP	Photochemical oxidation potential
RIVM	Rijksinstituut voor Volksgezondheid en Milieu
RVB	Rijksvastgoedbedrijf
SBK	Stichting Bouwkwiteit
SETAC	Society of Environmental Toxicology and Chemistry
S-LCA	Social Life-cycle Analysis
SO_x	Sulfur oxides
STEP	Standard for the Exchange of Product model data
TETP	Terrestrial ecotoxicity potential
tkm	Tonne-kilometer
UK	United Kingdom
US	United States

LIST OF FIGURES

FIGURE 1 - RESEARCH MODEL DESIGN	7
FIGURE 2 - THE THREE PILLARS OF LIFE-CYCLE SUSTAINABILITY ASSESSMENT	10
FIGURE 3 - SCHEME OF INFORMATION MODULES FOR DIFFERENT STAGES OF EVALUATION OF A BUILDING	14
FIGURE 4 - RELATIONSHIP BETWEEN DESIGN STAGES AND STUDY TYPES (BASED ON MEEX ET AL. (2018))	15
FIGURE 5 - LCA FRAMEWORK BASED ON ISO 14040 (INTERNATIONAL STANDARD ORGANIZATION, 1997).....	16
FIGURE 6 - A DECISION TREE BASED ON THE THREE PILLARS OF LCSA (ADOPTED FROM BUCHERT ET AL. (2015)).....	21
FIGURE 7 - A NETWORK GRAPH BASED ON THE DIFFERENT VECTORS AND EDGES	22
FIGURE 8 - EXPRESS DIAGRAM SHOWING A PART OF THE IFC-HIERARCHY (ADAPTED FROM DIMYADI ET AL. (2008)) ..	26
FIGURE 9 - ENVIRONMENTAL-ECONOMIC PERFORMANCE ASSESSMENT (AHMED & TSAVDARIDIS, 2018).....	27
FIGURE 10 - CUMULATIVE LCC AND LCA OUTCOME OVER 100 YEARS (RISTIMÄKI ET AL. (2013)).....	28
FIGURE 11 - FRAMEWORK FOR INTEGRATING LCA AND LCC (ADAPTED FROM PETRILLO ET AL. (2016))	29
FIGURE 12 - ENVIRONMENTAL AND ECONOMIC IMPACT ASSESSMENT MATRIX (SAVINO ET AL. (2017))	29
FIGURE 13 - INPUT-PROCESSING-OUTPUT METHOD (ADAPTED FROM NEGELE ET AL. (1999))	32
FIGURE 14 - LIST OF REQUIREMENTS FOR CALCULATING LCA AND LCC BASED ON IFC DATA	42
FIGURE 15 - SYSTEM'S ARCHITECTURE OF THE APPLICATION	44
FIGURE 16 - WORKFLOW OF THE DIFFERENT MODULES OF THE APPLICATION	44
FIGURE 17 - SNIPPET OF BASEVALUES.PY SCRIPT BASED ON A SELECTION OF DECISION CRITERIA	46
FIGURE 18 - TABS IN THE DATABASE CONSISTING OF LCA DATA	49
FIGURE 19 - TABS IN THE DATABASE CONSISTING OF LCC DATA	50
FIGURE 20 - ZOOMED VIEW OF THE SYSTEM'S ARCHITECTURE: COMBINING BoQ-CONTENT WITH LCA AND LCC DATA.	51
FIGURE 21 - GRAPHICAL USER INTERFACE OF LEICAS	54
FIGURE 22 - SNIPPET OF .XLSX-FILE BASED ON THE IFC-MODEL.....	56
FIGURE 23 - EXAMPLE OF A BAR CHART BASED ON THE GENERIC TYPE OF CALCULATION.....	56
FIGURE 24 - EXAMPLE OF A BAR CHART BASED ON THE SPECIFIC TYPE OF CALCULATION	57
FIGURE 25 - SNIPPETS FROM CONTOURING OF THE APARTMENT BUILDING	60
FIGURE 26 - SNIPPET WITH RESULTS OF THE SPECIFIC SCENARIO BASED ON BUILDING SYSTEM "ALL"	63
FIGURE 27 - SNIPPET WITH RESULTS OF THE GENERIC SCENARIO BASED ON BUILDING SYSTEM "ALL"	169
FIGURE 28 - SNIPPET WITH RESULTS OF THE SPECIFIC SCENARIO BASED ON BUILDING SYSTEM "ALL"	169
FIGURE 29 - SNIPPET WITH RESULTS OF THE GENERIC SCENARIO BASED ON BUILDING SYSTEM "BRICKS & CONCRETE"	170
FIGURE 30 - SNIPPET WITH RESULTS OF THE SPECIFIC SCENARIO BASED ON BUILDING SYSTEM "BRICKS & CONCRETE"	170
FIGURE 31 - SNIPPET WITH RESULTS OF THE GENERIC SCENARIO BASED ON BUILDING SYSTEM "WOOD"	171
FIGURE 32 - SNIPPET WITH RESULTS OF THE SPECIFIC SCENARIO BASED ON BUILDING SYSTEM "WOOD"	171

LIST OF TABLES

TABLE 1 - LIST OF INDICATORS FROM EN 15978 STANDARD (CURRAN, 2006; CAPAZ & SEABRA, 2016)	18
TABLE 2 - ENVIRONMENTAL IMPACT CATEGORIES WITH THEIR SHADOW COSTS FROM THE MPG.....	19
TABLE 3 - LIST OF LCA AND LCC STUDIES INTEGRATED WITH BIM (ADAPTED FROM LLATAS ET AL. (2019)).....	31
TABLE 4 - CHALLENGES OF INTEGRATING LCA, LCC AND BIM.....	34
TABLE 5 - PYTHON PACKAGES THAT HAVE BEEN USED FOR THE DEVELOPMENT OF THE APPLICATION	43
TABLE 6 - EXAMPLE OF CHANGES THROUGH THE ELECTION OF A TYPE OF BUILDING SYSTEM	53
TABLE 7 - FIELD OF IMPROVEMENTS	59
TABLE 8 - RESULTS OF THE LEICAS-APPLICATION, OPTIMISING THE BUILDING COMPONENT OF THE <i>SPECIFIC SCENARIO</i> .	63
TABLE 9 - CHANGING THE <i>GENERIC</i> SCENARIO BUILDING ASSEMBLY CODES TO THE OPTIMISED ASSEMBLY CODES.....	63
TABLE 10 - COMPARING THE OPTIMISED <i>GENERIC SCENARIO</i> RESULTS OF THE ENVIRONMENTAL IMPACTS	64
TABLE 11 - COMPARING THE OPTIMISED <i>GENERIC SCENARIO</i> RESULTS OF THE ECONOMIC IMPACTS	65
TABLE 12 - SENSITIVITY ANALYSIS COMPARISON OF THE ENVIRONMENTAL IMPACTS IN DIFFERENT APPLICATIONS	66
TABLE 13 - SENSITIVITY ANALYSIS COMPARISON OF THE ECONOMIC IMPACTS IN DIFFERENT APPLICATIONS	66

LIST OF EQUATIONS

EQUATION 1 - OVERALL ENVIRONMENTAL CALCULATION	20
EQUATION 2 - OVERALL SHADOW COSTS ENVIRONMENTAL IMPACT	20
EQUATION 3 - NPV-METHOD	20
EQUATION 4 - COST CALCULATION FOR MANUFACTURING AND CONSTRUCTION PHASE	20
EQUATION 5 - OVERALL SCALING FACTOR OF A PRODUCT COMPONENT	85
EQUATION 6 - SCALING FACTOR OF A PRODUCT THAT IS (RE)SCALABLE INTO TWO DIMENSIONS.....	85
EQUATION 7 - SCALING FACTOR OF A PRODUCT THAT IS (RE)SCALABLE INTO TWO DIMENSIONS.....	86
EQUATION 8 - NON-SCALABLE COMPONENT OF DIMENSION1 OF A PRODUCT	86
EQUATION 9 - NON-SCALABLE COMPONENT OF DIMENSION2 OF A PRODUCT	86
EQUATION 10 - ENVIRONMENTAL IMPACT OF A PRODUCT COMPONENT PER FUNCTIONAL UNIT	87
EQUATION 11 - ENVIRONMENTAL IMPACT OF THE COMBINED PROFILE FOR WASTE PROCESSING	87
EQUATION 12 - LIFESPAN OF A PRODUCT IN THE BUILDING	88
EQUATION 13 - LIFETIME FACTOR FOR A PRODUCT COMPONENT OVER THE ENTIRE LIFESPAN OF A BUILDING	88
EQUATION 14 - ENVIRONMENTAL IMPACT OF A PRODUCT COMPONENT PER IMPACT CATEGORY	88
EQUATION 15 - ENVIRONMENTAL IMPACT OF A PRODUCT PER IMPACT CATEGORY.....	89
EQUATION 16 - ENVIRONMENTAL IMPACT OF A PRODUCT OVER THE LIFESPAN OF A BUILDING PER IMPACT CATEGORY ..	89
EQUATION 17 - SHADOW COSTS OF A PRODUCT OF ONE IMPACT CATEGORY	89

1 INTRODUCTION

Over the last decades, carbon emissions have rapidly increased and are expected to grow continuously in the upcoming years. Furthermore, the global mean temperature has already increased by approximately 0.90 degrees due to global warming and is also expected to increase significantly, affecting the earth's life-support systems. Besides global warming, other environmental problems such as rising sea levels, increasing water, soil and air pollution will have enormous effects on human health, as well as natural eco-systems (Weisheng, 2017). In addition to the environmental problems experienced, it can be notified that an increase of scarcity in raw materials has emerged in the past centuries. The consumption of raw materials not only has a negative impact on the environment and its natural resources but also has a significant contribution to the increase of energy consumption and greenhouse gas emissions. To be more precise, the AEC-industry accounts for more than 30% of greenhouse gas (GHG) emissions per year globally (SER, 2016). Due to the explosive increase in demand for the utilisation of raw materials during the past century, the earth's life-support systems are threatened. According to the Sociaal Economische Raad (SER, 2016), an increase in the utilisation of materials, minerals and fossil fuels has emerged; the world population is using 34 times more materials, 27 times more minerals, and 12 times more fossil fuels since the last century. The demand for raw materials is expected to increase rapidly due to global population growth, the increase of new technologies that require specific raw materials, and the fast-growing middle-class economy in emerging economies (Bribián, Capilla, & Usón, 2011). An urgent need to diminish these problems is fundamental to preserve the earth's life-support systems.

In this chapter, the research setup will be explained. First, a description of the research context will be given. Next, the problem statement, the research objectives, the research questions, as well as the research design, will be elaborated on further. This chapter will end with a reading guide that clarifies the structure of this thesis.

1.1 RESEARCH CONTEXT

There is an urgent need for a transition from our current systems to more sustainable socio-technical systems (Markard, Raven, & Truffer, 2012). However, the current Dutch economy is still based on a linear system with a take-make-dispose model, and it can be notified that there is a slight transition toward a circular economy. The concept of a circular economy looks beyond the current take-make-dispose economy model (Ellen Macarthur Foundation, 2013), since circular economy tries to merge economic growth with the social progress and the sustainable use of natural resources while preserving the ecological balance for future generations (Drejeris & Kavolynas, 2014). Therefore, the interest of individuals, politicians, as well as companies, has been triggered in past years (Rijksoverheid, 2016).

Furthermore, a transition to a more circular economy is not an obstacle to further economic growth; a circular economy may even provide opportunities for Europe, as well as for the Netherlands (Ellen MacArthur Foundation, 2015). For example, it would allow Europe to rise the challenge of global pressure on resources and insecurity of material supply. However, at this moment in time, the AEC-industry in the Netherlands still uses large amounts of raw materials. According to Rijkswaterstaat and RIVM (2015), more than half of all raw materials in the Netherlands are being used in the AEC-industry. As the AEC-industry still puts extreme pressure on the natural environment, this sector must adopt a strategical approach covering several features such as reuse and recycling of materials, improved use of materials, emissions control and energy saving (Ramesh, Prakash, & Shukla, 2010). Uncertainties over the limitations of raw materials and energy resources started the interest in discovering approaches to embody energy use and to extend future resource supplies.

Life-cycle Thinking (LCT) is an approach that is gaining a prominent role in governmental policies, as well as in day-to-day thinking by individuals. LCT in the built environment implies representing the range of sustainable effects of a material or product throughout the whole lifecycle. Life-Cycle Assessment (LCA) is one of the methodologies that makes LCT operational. LCA is the study of resource consumption, emissions and their impacts. According to Filimonau (2016), LCA can be seen as *“a reliable, scientifically-grounded support tool for environmental management and decision-making across different sectors of the global economy”*. LCA is considered as an excellent way of assessing the potential environmental impacts of products and materials throughout their life-cycle (European Commission, 2003).

Despite the accuracy and consistency of LCA, the direct application of this method can be challenging (Filimonau, 2016). LCA studies need a high standard of information, particularly for the assessment of the embodied impacts of building materials (Soust-Verdaguer, Llatas, & García-Martínez, 2016). Therefore, LCA studies are mostly performed during the completion of the building and when all the information is clear, which limits the application of LCA in being only descriptive rather than providing feedback to improve the design (Jalaei & Jrade, 2015; Soust-Verdaguer, Llatas, & García-Martínez, 2017). This way of working is inefficient, since modifying the design to achieve a high standard of environmental performance criteria often results in backtracking (Schlueter & Thesseling, 2009). Besides the environmental evaluation, economic LCT of products is an essential factor within the AEC-industry. Although LCA focusses on the evaluation of environmental performance, LCC compares the cost-effectiveness of alternative investments from an economic perspective (Norris, 2001(a)). In other words, LCA follows the physical life-cycle of a product or material, while LCC follows its

marketing life-cycle. The development and implementation of LCA and LCC within the AEC-industry have been considerably slower compared to other industries due to the lack of interest of utilising LCT (Khasreen, Banfill, & Menzies, 2009). In contrary, related to the high environmental impacts of the AEC-industry on the environment combined with the economic boundaries of building projects, it can be stated that the importance of LCA and LCC within the AEC-industry has increased in the past decades (Khasreen, Banfill, & Menzies, 2009; Cabeza, Rincón, Vilariño, Pérez, & Castell, 2014; Anand & Amor, 2017). By only looking at the LCA methodology, many researchers stated that this methodology could directly be applied during the development of building design (Curran, 2013; Macombe, Leskinen, Feschet, & Antikainen, 2013; Soust-Verdaguer, Llatas, & García-Martínez, 2016). Therefore, integrating the approaches of LCA and LCC with each other could be beneficial for the development of a building design contemplating the environmental and economic principles of a project.

Still, the AEC-industry has been facing difficulties by assessing the environmental impacts combined with economic impacts of buildings. These difficulties occur due to uncertainties of design, for example, by the determination of the lifespan and the uniqueness of a building design during its entire lifecycle (Khasreen, Banfill, & Menzies, 2009). These uncertainties have a significant influence on the validity of designs since they are usually assumption based, especially during the early design stages. Based on their skills and knowledge, designers have to make well-considered decisions in order to create a design (Khasreen, Banfill, & Menzies, 2009; Anand & Amor, 2017). Therefore, it is essential to simplify the application of LCA and LCC approaches to develop a building (Soust-Verdaguer, Llatas, & García-Martínez, 2016).

Building Information Modelling (BIM) can have a supporting function, as it facilitates data that can be used for the application of LCA and LCC in the AEC-industry, mainly in the early design stages. During these early design stages, the potential for examining various alternatives, implementing changes, improving environmental performances and diminishing costs is the highest (Anand & Amor, 2017). BIM can give timely information about whole life environmental and economic issues as the design changes and develops, and it can also support in the decision-making process at the beginning periods of the plan (Azhar & Brown, 2009; Ajayi S. O., 2015; Anand & Amor, 2017; Santos, Costa, Silvestre, & Pyl, 2019). Because BIM takes multi-disciplinary information to be covered by one model into consideration, it provides an opportunity for environmental and economic assessment techniques to be incorporated in the design process, such as the LCA and LCC (Jrade & Jalaei, 2013). According to Azhar et al. (2008), BIM is an integrated process that involves collaboratively developing and using a computer-generated parametric model of a building, which is needed in order to facilitate life-cycle management of a building. Parametric modelling is a way to capture design form and functionality, which is achieved by using parameters and rules (Eastman, Teicholz, Sacks, Liston, & K., 2011). Bilal et al. (2016) suggest that parametric representation ensures preserving the form and functionality of designs in response to their contextual change. The development of BIM applications as well as international data exchange schemas such as the Industry Foundation Classes (IFC) could facilitate the storage and easy access of information for processing (Akbarnezhad, Ong, & Chandra, 2014).

In more recent years, the built environment has witnessed an increasing interest in the use of BIM in combination with sustainability principles during the design and construction stages of projects (Jalaei & Jrade, 2015). Various researches emphasise the importance of LCA and LCC

within a BIM environment in order to facilitate efficient decision-making, particularly in early design. This importance is because the decisions within these stages have a significant influence on the environmental impacts in other life-cycle phases of a building (Akbarnezhad, Ong, & Chandra, 2014; Ajayi S. O., 2015; Akinade, et al., 2015; Jalaei & Jrade, 2015; Santos, Costa, Silvestre, & Pyl, 2019; van Gemert, 2019). By promoting the potentials of the circular value of materials or products in the early design stages, the environmental and economic impact of buildings can be reduced (Akinade, et al., 2017). A solid combination between LCA and LCC with BIM, especially IFC, is key to ensure stimulation of the decision-making process at early design stages because information entities within BIM can support decision-makers to estimate environmental and economic impacts more accurate (Zhiliang, Zhenhua, Wu, & Zhe, 2011). Bank et al. (2010) mention that this combination is necessary in order to assess the different natural effects of plan and material choices more accurate. Therefore, a more holistic approach to assess the embodied impact of building materials is needed to get an integrated process between BIM with LCA and LCC.

It is acknowledged by various authors that further development in the field of the development of approaches toward the effective integration of LCA, LCC and BIM is necessary to implement LCA and LCC within the AEC-industry (Akbarnezhad, Ong, & Chandra, 2014; Akinade, et al., 2015; Verberne, 2016; Petrillo, et al., 2016; Santos, Costa, Silvestre, & Pyl, 2019; van Gemert, 2019). Even though there are already some IFC-applications that implement these LCA or LCC approaches, research on the possibilities of IFC to support data exchanges between LCA and LCC is still in its early stages. By researching these data exchange possibilities of IFC, the integration of LCA with LCC can be simplified.

Although some work has been done to enable BIM for environmental and economic impact assessment of building designs (Liu, Meng, & Tam, 2015; Alwan, Jones, & Holgate, 2017; Akanbi, et al., 2018), the work is still lacking in terms of useful feedback. With regards to the provided feedback, the feedback mainly provides end-results of environmental or economic impact. Based on these end-results, designers have to make decisions related to the environmental and economic issues of a building design. However, only skilled designers that know how to interpret the results know how to improve the designs environmentally as well as economically. Pereira and Fabricio (2018) proclaim that providing useful feedback based on LCA related design decisions in the design is key for improving the design environmentally in early design stages. By having useful feedback, designers are enabled to make decisions faster and more well-considered, which will positively affect the way of decision-making.

Future development of BIM-based environmental and economic impact assessment for evaluating the environmental and economic sustainability of buildings is necessary. BIM-based LCA and LCC with a built-in feedback feature will provide designers with the opportunity to make considered environmental and economic decisions of a design. According to Van Gemert (2019), the chances of human failure are reduced by the availability of feedback features during an LCA-study. This reduction of human failures is a result of design changes based on the preferences and the ideologies of involved parties during the design stages such as architects, clients and project managers. Furthermore, this will positively affect the communication and interpretation of messages between these stakeholders. Feedback within an LCA and LCC integrated BIM-environment is key to speed up and improve the quality of the process of considering environmental and economic decision-making.

1.2 PROBLEM STATEMENT

The AEC-industry is putting more emphasis on the development of BIM-based LCA and LCC possibilities (Anand & Amor, 2017; Akanbi, et al., 2018). By promoting the decision-making process at the early design stages using the integration of BIM with LCT approaches, designers can evaluate various environmental and economic impacts of their design and assess the different natural effects of material choices more accurate (Bank, McCarthy, Thompson, & Menassa, 2010). Therefore, a holistic approach is needed in order to provide an opportunity for the establishment of assessing the embodied impact of building materials with regards to their environmental and economic impact.

Recent research has shown the importance of LCA and LCC within a BIM-based environment to facilitate efficient decision-making, particularly in early design stages (Akbarnezhad, Ong, & Chandra, 2014; Ajayi S. O., 2015; Akinade, et al., 2015; Jalaei & Jrade, 2015; Petrillo, et al., 2016; Santos, Costa, Silvestre, & Pyl, 2019; van Gemert, 2019). Besides, several types of research still acknowledge the necessity of further development in the field of providing feedback based on the environmental and economic impact of design (Díaz & Antón, 2014; Akanbi, et al., 2018; Bueno, Pereira, & Fabricio, 2018; van Gemert, 2019). Nowadays, integrated environmental and economic impact assessment and BIM applications, which are developed by software companies, only address interoperability among their products. Furthermore, they serve as plugins for specific software applications and are lacking feedback opportunities. Since the compatibility of such applications toward other applications is deficient (Soust-Verdaguer, Llatas, & García-Martínez, 2017), the need to develop new applications based on open BIM schemes, especially with the use of the IFC, is necessary. The potential of a feedback feature will provide designers knowledge in how to make considered environmental and economic decisions within a design.

1.3 RESEARCH OBJECTIVES

Based on the societal development and the identified knowledge gap, this research explores the potentials of feedback integrated with BIM-standards, considering the environmental and economic impact of building materials. For the calculation of the environmental impact, the Dutch standard MPG is used. MPG is a calculation method for determining the environmental performance of buildings and civil engineering works according to the LCA-method. In other words, the methodology of MPG provides a comprehensive analysis of different environmental aspects. For the LCC-approach, the information that can be retrieved from an IFC model, only the acquisition costs of the building have been used in order to determine the economic impact of the design. This approach means that only the costs of the materials and the construction of these materials and products in a building are considered. The economic impact calculations are based on the economic data available for materials combined with the information of the BIM-model.

Given the time frame of six months for this research, the focus of the application lies within the development of an application that performs an environmental and economic impact assessment considering the wall and floor elements. This consideration was made with regards to the environmental impact assessment since the embodied impacts of wall and floor elements are straightforward to calculate in comparison to other building elements. Building parts that have the highest contribution to the environmental impact are walls, floors and installations; they can take up to 60% to 80% of the results (Bouwen met Staal, 2019; RVO,

2019). In general, installations are not specified or available during the early design stages. As a base for the application, the decision is made only to incorporate wall and floor elements.

The primary objective of this study is to show that a BIM-based assessment application, existing of embodied impacts can guide the decision-making process of designers by (1) generating an automated environmental and economic impact assessment, and (2) providing feedback to the designers. By applying such an approach, the importance of integrating LCA and LCC within BIM to assess the environmental and economic impact of the embodied impacts of building elements is emphasised. Based on the objective, the expected result is a literature research which contains the answer to the main research question and Python-scripts that will provide an application which can be used for the environmental and economic impact assessment of a BIM-model.

1.4 RESEARCH QUESTION

In order to address the problems within the problem definition and the desired research objectives, the following main question was drawn up:

How can the integration of *Life-Cycle Assessment* and *Life-Cycle Costing* with *Building Information Modelling* support designers in decision-making by enabling an automated *Environmental and Economic Impact Assessment* during the early design stages?

The main question will be answered by answering the following sub-questions:

- SQ1 To what extent are LCA and LCC currently applied within the AEC-industry?**
- SQ2 In what way can BIM be utilised to support designers in making environmental and economic decisions based on the LCA and LCC methodology?**
- SQ3 Which different feedback mechanisms influence the decision-making of the environmental and economic impact of a building?**
- SQ4 What are the constraints of a BIM-based environmental and economic impact assessment application?**
- SQ5 To what extent is decision-making in early design stages enhanced by applying a BIM-based environmental and economic impact assessment application?**

1.4.1 RESEARCH LIMITATIONS

By keeping a keen eye on the feasibility of the research, the main focus of the application will be on the development of an application that performs an environmental and economic impact assessment with a feedback feature considering the wall and floor elements. Since embodied impacts of wall and floor elements are straightforward to calculate in comparison to other building elements, the choice for incorporating wall and floor elements in a proof-of-concept seems the most logical choice.

The quantification and calculations in the application for the environmental impact of the design will be established by using the MPG-method. Following the MPG-method, the categorisation of the different building elements is based on the environmental database, the *Nationale Milieu Database* (NMD). This database is mandatory to use for calculating the environmental impact of the MPG-method. With regards to the calculation of the economic impact of design and based on the information that can be retrieved from an IFC-model, only the acquisition costs of the materials can be calculated. Within these acquisition costs, the

costs of the materials and the construction of these materials in a building are considered. By linking the economic data for calculating the acquisition costs to the required environmental database, the application is enabled to make a combined impact assessment.

1.5 RESEARCH DESIGN

This research study is divided into four different stages, which can be seen in Figure 1: (i) explorative research; (ii) application design; (iii) verification; and (iv) reporting. In order to get a specific understanding of the content of the research stages, the following four sections will provide a more in-depth discussion of these four research stages.

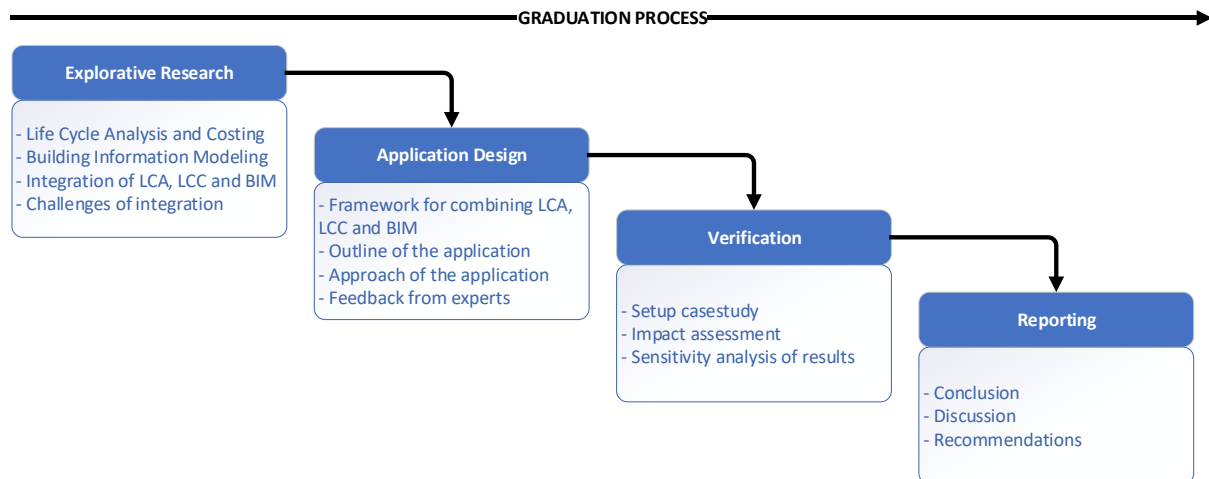


Figure 1 - Research Model Design

1.5.1 EXPLORATIVE RESEARCH

The explorative research stage consists of literature research, unstructured interviews and data gathering. Within the literature research, three main topics are explored; BIM, LCA and LCC. This literature research focusses on the current situation of LCA, LCC and BIM implementation and utilisation in the Dutch AEC-industry, especially the linkage between the subjects. By using the knowledge gathered in the literature research, the different limitations of LCA, LCC and BIM were combined and used to set a base for the development of the application and are validated by unstructured interviews. Secondly, considering the first literature research, research is done on the different required parameters of a BIM-model, specifically related to the environmental and economic impact and data interoperability. An essential goal of this literature research is the research into the appliance of BIM, in which the focus will lie within the integration of LCA and LCC within BIM, potentially with feedback opportunities.

1.5.2 APPLICATION DESIGN

The application design stage entails the development process of an application that enables the user to get insight into the assessment of the embodied environmental and economic impact of building elements. The main challenge in the model design stage lies in linking the data from a BIM-model to an application that provides an automatic overview of the related attributes of building components. The incorporated environmental method of MPG, which is going to be used for the environmental calculations of the application, needs to include a uniform calculation performance of the environmental impact of building elements. Furthermore, the application must be in line with the LCT way of thinking; it requires the recording of the life-cycle sustainability performance of the design process.

In order to develop the application, the programming language that is going to be used is Python. Python can be considered as a standard syntax and allows third-party packages to be implemented in order to extend the existing functionality of the Python language. With regards to the input of the data and because the embodied impact of the BIM-model must be calculated, a combined LCA and LCC database is needed. The input of the LCA database, which is based upon the MPG-method, and the LCC database relates to the environmental and economic characteristics of building products and their predefined material composition. The added value of the database is that it provides structured data on environmental and economic related attributes of building products and materials in the form of a machine-readable file format. Such a database is necessary in order to link and compare the values of a BIM-model with shared values of the database.

1.5.3 VERIFICATION

The verification stage consists of testing the application with a case study. With regards to the content of the verification of the case study, the features of the application are tested and examined through an apartment building. The main objective of this case study is to demonstrate how a BIM-based LCA and LCC application can be used in order to execute an environmental and economic impact assessment of the whole life embodied impacts of building elements and how feedback can be provided that is used for the determination of choosing building materials of a building. This determination will be based on the present uncategorised building components in a design. In other words, the application provides calculations for generic, unclassified components and specific, classified components. The generic, unclassified components in a building design contain only the location and function of the components. The specific, classified components in a building design contain the location and function of the components like the generic components, but also contain information on the used materials for those components. This specification will enable a more precise calculation for all the specified components in building design.

During the development of the application, the proposed application is presented to some experts within the graduation company. This verification is necessary in order to maintain the main focus of the application and to determine if the proposed application still has added value during the decision-making in early design stages.

1.5.4 REPORTING

The reporting stage entails the finalisation of the research report. In this final stage of the research, conclusions are drawn, the research question with its sub-questions are answered, the overall research is discussed, and recommendations for further research are provided. The graduation report is updated during the entire graduation period. Therefore, the research report is close to completion when this phase of the research is reached.

1.6 READING GUIDE

This thesis starts by providing knowledge on the subjects LCA, LCC and BIM, which ultimately forms the literature research. This literature research is presented in chapter 2. Chapter 3 presents the development of the application, which consists of the elements of a *minimum viable product*, the content of the application and the system's architecture on how to use the application. Chapter 4 elaborates on the demonstration of different features of the application through a case study of an apartment building. To conclude the content of the thesis, a conclusion will be drawn, and recommendations of future research are provided in chapter 5.

2 LINKING THE DOTS: LCA, LCC AND BIM

In the past decades, the interest in the development of methods to understand and reduce environmental impacts have been aroused in the AEC-industry. One of these methods that have been used in LCA. On this day, LCA is a widely applied method to support policies and performance-based regulation (Guinée, 2016). Besides LCA, the interest to include economic (LCC) and social (S-LCA) subjects related to the life-cycle of a building gained in the AEC-industry. These three components form the three-pillar model of sustainability, called life-cycle sustainability assessment (LCSA). LCSA enables practitioners to organise complex environmental, economic and social data in a more structured form (Santos, Costa, Silvestre, & Pyl, 2019).

In more recent years, several studies focus on the integration of sustainability in the AEC-industry, mostly regarding the environmental (LCA) and economic (LCC) dimensions. In this field, one of the recent trends is the integration of Building Information Modelling (BIM) with LCA and LCC (Guinée, 2016; Anand & Amor, 2017; Nwodo, Anumba, & Asadi, 2017; Soust-Verdaguer, Llatas, & García-Martínez, 2017; Zuo, et al., 2017; Iacovidou, Purnell, & Lim, 2018). BIM can be used as an effective way of modelling and managing the project's information while enabling design changes to be made in a fast and effortless way (Iacovidou, Purnell, & Lim, 2018). By using the integration of BIM with the LCSA methodology, designers can evaluate various environmental impacts and economic variations of their design more accurate (Bank, McCarthy, Thompson, & Menassa, 2010). Recent research has shown the importance of LCA and LCC within a BIM-based environment to facilitate efficient decision-making, particularly in early design stages (Akbarnezhad, Ong, & Chandra, 2014; Ajayi S. O., 2015; Akinade, et al., 2015; Jalaei & Jrade, 2015; van Gemert, 2019; Santos, Costa, Silvestre, & Pyl, 2019). However, although research has been done on the importance of integration of LCA with LCC and integrating BIM with LCA, less research is done on the integration of the three components with each other.

The theoretical background of this research is explained in this chapter. The following sections exclusively focus on three subjects: LCA, LCC, and BIM. First, the general idea of LCA and LCC considering the LCSA, as well as BIM, will be researched. Second, the integration of LCA and LCC with and without BIM is examined. Finally, the challenges and potentials of the integration of BIM with LCA and LCC are reviewed which form the starting point for the development of a BIM-based LCA and LCC application.

2.1 LIFE-CYCLE ASSESSMENT AND LIFE-CYCLE COSTING

The most researched subject of LCSA is LCA. LCA can be described as a method for systematically analysing the environmental performance of products and processes considering their entire life-cycle (Santos, Costa, Silvestre, & Pyl, 2019). LCA could provide a better understanding of the potential environmental impact on the decision-making process (Shi, Liu, Zhang, & Jiang, 2015). However, LCA cannot determine which product or process is the most cost-effective. Different researches propose that in order to make an LCSA, LCA needs to be combined with economic and social aspects (Guinée, 2010; Shi, Liu, Zhang, & Jiang, 2015; Fauzi, Lavoie, Sorelli, Heidari, & Amor, 2019). Furthermore, Hunkeler and Rebitzer (2005) mention that, with regards to products and materials, the assessment of sustainability will require approaches for environmental, economic and social evaluation. In order to understand LCSA content-wise, the three pillars of LCSA will be discussed.

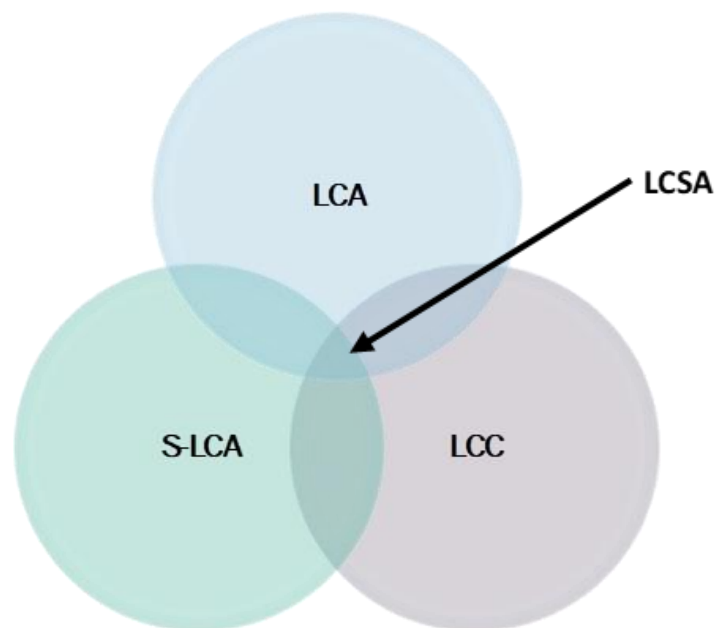


Figure 2 - The three pillars of life-cycle Sustainability Assessment

2.1.1 LIFE-CYCLE ASSESSMENT

One of the first publications on LCA, written by Harold Smith in 1963, provided insight into the calculations of cumulative energy requirements to produce chemical intermediates and products. In the late sixties and early seventies, LCA was used as tool which was developed by companies for their resource management, mainly for the environmental assessment of single product development processes in industries such as consumer product design, equipment manufacturing and automotive design (Curran, 1993; Hunt & Franklin, 1996; Basbagill, Flager, Lepech, & Fischer, 2013). Among the first, Coca Cola produced an analysis encompassing a more comprehensive range of environmental impacts (Guinée, et al., 2011).

Unfortunately, due to the commercial sensitivity of the content, many LCA and LCC studies have not been published. However, these initial studies formed the foundation for the conditions of environmental and economic assessments that are still known and used today. With regards to the LCA, the ideas of so-called eco-balances revealed the fast-growing waste problem as a potential resource in Europe during the mid-eighties. According to Buonocore et al. (2009), it was not until the late eighties that an increased interest in LCA became more prominent in other industries such as the AEC-industry. After the publication of the

Brundtland report by the United Nations in 1987, the importance of sustainable development became more explicit, which simultaneously led to an increase of academic interest in LCA (United Nations, 1987). This importance was because LCA was getting acknowledged as a useful tool to calculate impact, especially with regards to sustainability, across a range of issues. The first mentioning of the term “Life-Cycle Assessment” was in the report on the Smugglers Notch (Vermont) workshop in 1990, providing a methodology that is still used to this day (Klöpffer W. , 2006). The idea expanded into the determination of environmental impacts in the seventies with resource and environmental profile analysis in the US.

However, LCA studies were frequently executed using a variety of methods (Buonocore, et al., 2009). Besides, without having a uniform theoretical framework, LCA was prevented from becoming a generally accepted and applied analytical instrument (Fauzi, Lavoie, Sorelli, Heidari, & Amor, 2019). Due to the lack of standardisation, the Society of Environmental Toxicology and Chemistry (SETAC) presented scales of magnitude for the establishment and measurement of environmental impacts of single products; the SETAC-standards (McManus & Taylor, 2015). This development meant that LCA as a tool could be used for regulatory objectives. Therefore, SETAC-standards were amended to ISO standards in the late 1990s, forming the ISO 14040 for LCA. Through the years, the following standards were used to conduct an LCA -study:

- ISO 14040:1997/2006 on principles and framework for LCA;
- ISO 14041:1998 on goal and scope definition and inventory analysis;
- ISO 14042:2000 on impact assessment;
- ISO 14043:2000 on interpretation;
- ISO 14044:2006 on requirements and guidelines for performing LCC-analyses.

It must be noted that the ISO 14041, ISO 14042 and ISO 14043 are withdrawn by ISO (ISO, 2006). These standards are revised by both ISO 14040 as well as ISO 14044, which are therefore currently used as the LCA standards.

2.1.2 LIFE-CYCLE COSTING

In the 1960s, the U.S. Department of Defense carried out a life-cycle analysis method for the procurement of military equipment that was categorised as high-cost products; the start of LCC (Fauzi, Lavoie, Sorelli, Heidari, & Amor, 2019). The general concept of LCC was first defined by Blanchard (1978), by mentioning that LCC is an approach that intends to design in such a way that costs are minimised over the entire life-cycle of a system or product. Concerning the economic sustainability principle, guidance is needed for the assessing of the total cost performance of an asset over time, including acquisition, operation, maintenance, and disposal costs. Therefore, Life-Cycle Costing is fundamental to the current drive to achieve better value for money from buildings (Langdon, 2007).

Whereas LCA evaluates the relative environmental performance of alternative production systems for providing the same function, LCC compares the cost-effectiveness of alternative investments or business decisions from the perspective of an economic decision-maker (Norris, 2001(a)). Besides, LCA and LCC follow a different strategy related to the life-cycle. It can be stated that LCA follows the physical life-cycle, which means a cradle to grave approach for the life-cycle of, for example, a product or material. On the other hand, LCC follows, in general, a marketing life-cycle approach of a product or material. This approach relates to the

life-cycle from a product; from the product development to the end market. With regards to the ISO 14040 related to the LCA-approach, the ISO 15686 defines LCC as a method for systematic evaluation of life-cycle costs over the analysis period. Especially ISO 15686-5, which covers the service life planning, the long-term understanding of building components and materials (Ahmed & Tsavdaridis, 2018). Through these standards, Life-Cycle Costing can be used as a comparison tool to understand the life-cycle costs of different alternatives of a design as part of a decision-making process.

However, LCC as a method for the determination of Life-Cycle Costing in the Dutch AEC-industry is still lacking in terms lack of LCC-calculation standards. Although researches propose methods for the calculation of LCC (Swarr, et al., 2011; Heijungs, Settanni, & Guinée, 2013; Santos, Costa, Silvestre, & Pyl, 2019), the lack of standardisation methods of calculating LCC does not oblige the Dutch AEC-industry to utilise LCC as a method. By requiring the utilisation of LCC as a method in the AEC-industry establishes a sustainable way of thinking with regards to the economic dimension.

Another challenge LCC faces related to the implementation of LCC in the AEC-industry is that there is currently a lack in dedicated databases that provide benchmarks for the calculation of life-cycle costs (Caspeelee, Taerwe, & Frangopol, 2018). Besides, Caspeelee et al. (2018) proclaim that *“data in the public domain is often aggregated and incomplete”*. Besides, commercial LCC-databases are often linked to the country of origin, meaning that the used methods for calculating the economic life-cycle of design are based on local regulations of the country where the LCC-database is used. Besides, by looking at a linkage of the three pillars of LCSA, it can be stated that there is currently no database on the market that links the three pillars. A more generalised and interlinked database of LCA, LCC and S-LCA would be beneficial for the implementation of LCSA in the AEC-industry.

2.1.3 SOCIETAL LIFE-CYCLE ASSESSMENT

The newest addition to performing an LCSA is by taking social indicators into account (Life Cycle Initiative, 2013). S-LCA is a method that can be used to assess the social and sociological aspects of products and materials with regards to their actual and potential impacts along the life-cycle. The social benefits can be estimated by analysing the effects of the stakeholders at different levels, such as on local, national and global level. The standard social indicators measure the degree of societal values and to what extent life's goals can be reached (Barros Telles do Carmo, Margni, & Baptiste, 2016).

Unlike conventional LCA with relatively clear cause and effect chains, in S-LCA the cause and effect chains are challenging to correlate making it quite challenging to select appropriate indicators (Siebert, Bezama, O'Keeffe, & Thrän, 2016). To be more specific, S-LCA makes use of generic and site-specific data, it can either be quantitative, semi-quantitative or qualitative, and it complements the LCA and LCC, making it hard to define and select indicators related to the social and sociological aspects of products and materials (Life Cycle Initiative, 2013; Shi, Liu, Zhang, & Jiang, 2015). However, many social issues are not easily quantifiable in performance measurements (Jørgensen, Finkbeiner, Jørgensen, & Hauschild, 2010).

In contrast to environmental and economic aspects of sustainability assessment, a social assessment still lacks a broad consensus on adequate indicators or standardised methods

(Klöppfer W. , 2008; Hu, Kleijn, Bozhilova-Kisheva, & Di Maio, 2013; Guinée, 2016; Petrillo, et al., 2016). Besides, there are currently no standardised indicators established. Macombe et al. (2013) claim that the S-LCA method is not yet fully developed and that practitioners are not able to measure the social impacts, but only describe the social performance. Due to the wide variety of social indices and indicators applied in the literature (Siebert, Bezama, O'Keeffe, & Thrän, 2016; Barros Telles do Carmo, Margni, & Baptiste, 2016), it is difficult to perform a well-founded S-LCA. Therefore, within this research, the focus of the driving principles of sustainability will lie in the combination of LCA and LCC.

2.2 THE ROLE OF LIFE-CYCLE ASSESSMENT AND LIFE-CYCLE COSTING IN THE AEC-INDUSTRY

The development and implementation of LCA and LCC within the AEC-industry have been significantly slower compared to other industries due to reluctance of utilisation (Khasreen, Banfill, & Menzies, 2009). However, because of the high environmental impacts of the AEC-industry on the environment, it can be stated that the significance of LCA and LCC has increased in the past decades. Particularly around the beginning of the 21st century, the relevance of applying the LCA methodology was extensively described within a variety of articles (Khasreen, Banfill, & Menzies, 2009; Cabeza, Rincón, Vilariño, Pérez, & Castell, 2014; Anand & Amor, 2017). In general, the LCA methodology could directly be applied to the development of a building (Curran, 2013; Macombe, Leskinen, Feschet, & Antikainen, 2013; Soust-Verdaguer, Llatas, & García-Martínez, 2016).

However, in comparison to other industries, the AEC-industry has been facing difficulties by assessing the environmental and economic impacts of buildings. These difficulties are because of uncertainties in design, such as the long lifespan of a building compared to the shorter lifespan of certain materials and components and the uniqueness of building designs during its entire lifecycle (Khasreen, Banfill, & Menzies, 2009). These uncertainties have a significant influence on the validity of end-results since they are generally based upon assumptions. Designers have to make well-considered decisions based on their skills and knowledge for the development of design (Khasreen, Banfill, & Menzies, 2009; Anand & Amor, 2017).

Another challenge in utilising LCA and LCC by the AEC-industry lies within the complexity of LCA and LCC tools. Current assessment tools are considered to be quite complicated in terms of utilisation by designers or other parties without professional knowledge. In general, the tools consisting of the LCA and LCC calculation methods have been developed for a post-construction assessment of a building by which an LCA-expert utilises the tools. Therefore, the results of the tools are mainly applied for scientific research purposes and thus limits the comprehension by non-LCA and LCC experts during the development of a design. In other words, in general, the results cannot be interpreted by non-LCA and LCC experts, which results in the lack of implementation of LCA and LCC in the AEC-industry.

During the design process, the different parties involved in a project need to collaborate, which means they may have different interests in design. For example, designers pay more attention to the environmental impact of design, while project managers focus on the economic impact of design. Current tools cannot meet these different needs simultaneously. The need for a strategic approach is key to implement LCA and LCC in the AEC-industry.

2.2.1 BUILDING LIFE-CYCLE STAGES

For the performance and utilisation of LCA and LCC in the AEC-industry, various stages of a building's life-cycle can be taken into account. The definitions of the various stages of a buildings' life-cycle, mentioned in the EN 15978 (2011), are: (A) product stage; (B) construction stage; (C) use stages of the building fabric and operation; and (D) end-of-life-stage – see Figure 3 (Soust-Verdaguer, Llatas, & García-Martínez, 2016).

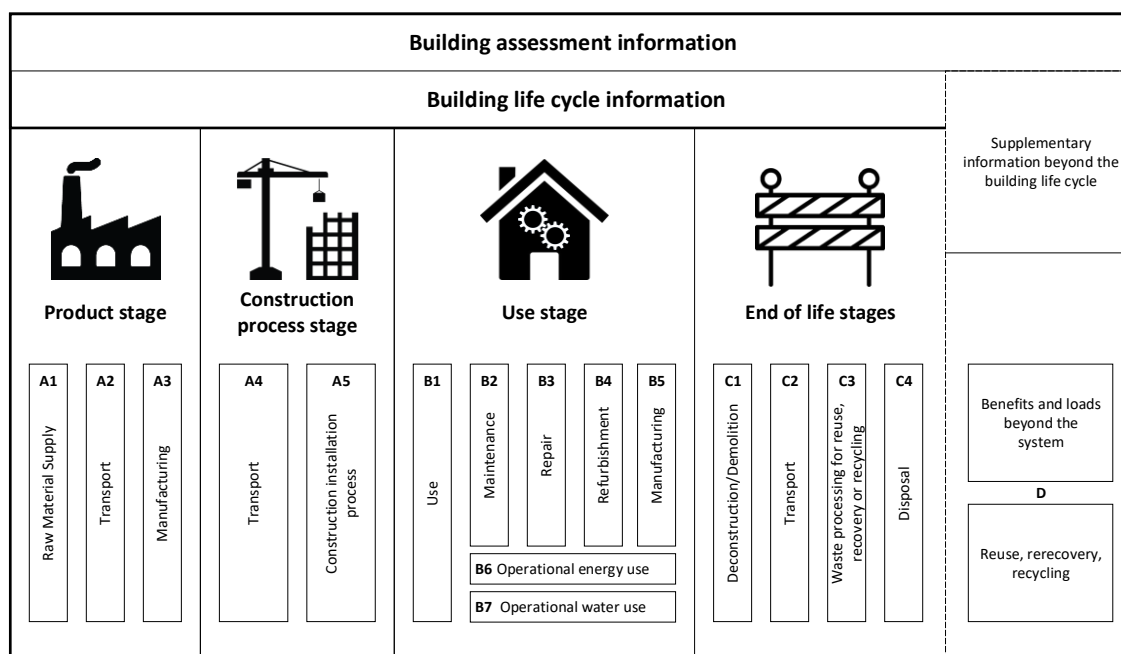


Figure 3 - Scheme of information modules for different stages of evaluation of a building

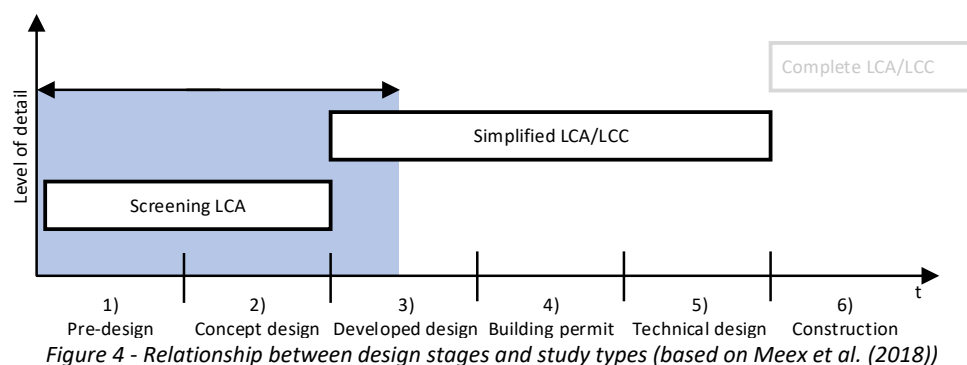
The product stage (A1-A3) involves the cradle-to-gate processes such as the extraction of resources, the transport of materials to manufacturing facilities and the manufacturing of building products. The construction stage (A4-A5) embodies all construction and installation activities that are related to the realisation of a building, including the transport of materials and products from factory to the manufacturer and from manufacturer to building site. The use stage (B1-B7) can be split into two different stages. The first scenario is the use of building fabric (B1-B5), which includes the use of construction products and services as well as scenarios for maintenance, repair, replacement and refurbishment. The second scenario is the use stage (B6-B7), which consists of the operational energy and water use of the product. The end-of-life stage (C1-C4) starts when the building's life is completed. Processes that may be considered are a deconstruction, transportation of discarded materials, waste processing for reuse, recovery or recycling and disposal. The final stage (D), gathers all embodied effects of reuse, recovery or recycling beyond the building's life-cycle. Through these different life stages, energy and materials are consumed, which result in the depletion natural resources and the pollution through harmful emissions, for example.

According to Wittstock et al. (2012), three different types of LCA studies, which can also be applied on LCC studies, can be conducted in the AEC-industry: (i) screening, (ii) simplified, and (iii) complete. During the *EeBGuide Project*, conducted by Wittstock et al. (2012) it was recognised that the use of the LCA studies in the AEC-industry could not be executed with the same level of detail as in other sectors because the application of an LCA-study within a building is complicated and time-consuming. In addition to the aforementioned LCA and LCC

studies, the European research project named 'ACADEMY' promoted a fourth type for the AEC-industry (ACADAMY, 2008, p. 24), which is based on the streamlined LCA approach proposed by SETAC in 1999 (Curran & Todd, 1999).

Within the early design stages, considering the pre-design and concept design stage, a screening study can be used for the initial assessment of the environmental impacts of buildings and products. However, since at these stages, not much data is available yet, performing a screening LCC-study is highly dissuaded. Therefore, instead of a screening LCC-study in the early stages, a similar simplified study is generally done in a more advanced stage when more data is available. This means that both LCA, as well as LCC studies, can be performed, using a simplified study approach. With regards to a complete study, a comprehensive assessment of the environmental impact and costs of the building or product can be performed, covering the entire life-cycle of the building or product. Therefore, a complete study is more suited for the advanced stages of a building project. The streamlined study can be described as an ad hoc version of a standard LCA approach in which the decision-maker selects the most suitable boundaries and categories for their environmental study. In order to execute an LCC-study in the early design stages, the streamlined approach can be considered the most optimal study to be used.

According to Santos et al. (2019), in the case that quantitative data of the system is not available, which is the case in early design stages, the streamlined study is recommended for the assessment of different products with regards to their life-cycle. Figure 4 provides an overview of the different LCA and LCC-studies different stages according to their level of detail. Since the LOD of the streamlined study is not formulated clearly, this approach cannot be included in Figure 4. To conclude, the different types of studies differ in terms of the completeness of the assessment, data representativeness, documentation and communication of results (van Gemert, 2019).



2.2.2 THE METHODOLOGICAL FRAMEWORK OF LIFE-CYCLE ASSESSMENT AND LIFE-CYCLE COSTING

Considering the international standard ISO 14040, the methodological framework of LCA studies consists of four stages which are interconnected in some way, as shown in Figure 5. The four stages include: (i) goal and scope definition; (ii) inventory analysis; (iii) life-cycle impact assessment (LCIA); and (iv) interpretation. According to Matthews, Hendrickson and Matthews (2014), none of the life-cycle stages is completed until the other stages are finished. As can be seen in Figure 5, the arrows represent an iterative process between different stages.

In 2008, SETAC-Europe published a book which described three different categories of LCC: (i) conventional LCC; (ii) environmental LCC; and (ii) societal LCC (Hunkeler, Lichtenvort, &

Rebitzer, 2008). Whereas conventional LCC focuses on internal costs directly associated with a product's life-cycle, environmental LCC goes even further with including external costs that are likely to be internalised in the future (Hunkeler, Lichtenwort, & Rebitzer, 2008). Societal LCC incorporates the costs that emerge from the side-effects of production with regards to people's lives and society. Generally, it has been agreed that environmental LCC is appropriate to be used alongside LCA since an environmental LCC analysis has a similar structure as an LCA (Krigsvoll, 2007; Chang, Neugebauer, Lehmann, Scheumann, & Finkbeiner, 2017).

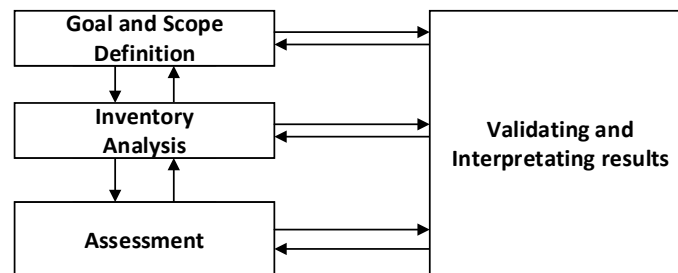


Figure 5 - LCA framework based on ISO 14040 (International Standard Organization, 1997)

Overall, the life-cycle and system boundaries need to be corresponding, but they do not have to be the same entirely since different processes may have different relevance for the environment and the costs (Ciroth & Franze, 2009). Therefore, from this point, when referred to an LCC-study, the environmental LCC is meant.

GOAL AND SCOPE DEFINITION

By defining the goal and scope of the LCA-study, the functional unit, system boundaries and data quality criteria are established (Cabeza, Rincón, Vilariño, Pérez, & Castell, 2014). The functional unit measures the performance of the output of a product system within the LCA-study. The system boundaries determine which unit processes are included within the LCA. These unit processes refer to the smallest elements of a product system for which data is collected during the *Inventory Analysis*. One of the main elements of setting the boundary is how to include certain life-cycle stages into the LCA-study, see Figure 5. The data quality criteria specify the required characteristics of data which is needed for the establishment of an LCA-study (International Standard Organization, 1997). Ten key categories should be considered when data quality is assessed: (i) time related coverage; (ii) geographical coverage; (iii) technology coverage; (iv) precision; (v) completeness; (vi) representativeness; (vii) consistency; (viii) reproducibility; (ix) sources of the data; and (x) uncertainty of the information (ISO, 2006). These categories are an excellent way to process the data. Practitioners of LCA must give their interpretation on how to address the different categories (Edelen & Ingwersen, 2016).

INVENTORY ANALYSIS

The integration of LCC with LCA must be done within the inventory analysis stage of the LCA process because of the input data inventorying as well as output data inventorying for individual processes and selected functional units takes place in this stage (Swarr, et al., 2011). As mentioned by Guinée (2002), life-cycle inventory analysis comprises the process of collecting and synthesising data on physical material and energy flows, considering both the input as an output of the data, in different stages of the products life-cycle. The flows can be described as the input flows that enter the system boundary from nature such as materials and energy, and output flows that leave the system boundary to nature such as products, waste and emissions (Pinto, 2013).

An inventory analysis follows a four-step approach: (i) development of a flow diagram of the evaluated processes; (ii) development of a data collection plan; (iii) the collecting of data and (iv) the processing of results (Curran, 2006). The first step is to map all input and output data of the system into a flow diagram. From that point, a data collection plan can be developed. Such a plan is an addition to the defined data quality criteria within the goal and scope definition stage. In order to define a data collection plan, the following key elements have to be included: (i) defining the data quality goals; (ii) identifying different data types and sources; (iii) identifying data quality indicators, and (iv) developing a data collection worksheet (Curran, 2006). Finally, inventory results are examined, processed and reported.

It is common for the performance of a life-cycle inventory analysis to make a distinction between two approaches: (1) consequential, and (2) attributional modelling. Both approaches are related to the functional unit, which can be described as a quantified performance of a product system for use as a reference unit, which includes both the physical as the functional components of a building. In general, a functional unit encompasses the aspects of function, quantity, duration and quality (Soust-Verdaguer, Llatas, & García-Martínez, 2016). The consequential approach is an approach in which activities in a product system are linked to include activities in the product system due to a change in demand for the functional unit (Ekvall, et al., 2016). The attributional approach, however, is an approach that links and subdivides unit processes of the system so that inputs and outputs of a production process are attributed to the functional unit of a product system (UNEP/SETAC, 2011). *Attributional LCA* aims to impute a share of the total environmental impact to a specific product system, while *Conceptual LCA* aims to denote the environmental impact that occurs through a change of the specific product system (Weidema, 2003). With regards to the ISO 14040 series, basic requirements are set for consequential LCA. According to Weidema (2003), consequential LCA is found to be the appropriate method with regards to LCA to reflect consequences of choosing between products. Although it must be stated that the significance of attributional LCA is not lesser than that from consequential LCA, the focus within this research shall lie within the consequential LCA since this approach falls within the scope of the research¹. Therefore, from this point, when referred to an LCA-study, the consequential modelling approach is meant.

IMPACT ASSESSMENT

In the impact assessment stage, the potential environmental impacts of various material and energy flows are allocated to different environmental impact categories in which the characterisation factor can be used to calculate the contribution of each environmental impact category (Finnveden, et al., 2009), such as climate change, human toxicity, and resource depletion. With regards to LCC, within the assessment stage, calculations are made on the potential costs. The main aim of the assessment stage is to establish a link between a product system with its potential environmental and economic impacts. There are different ways of calculating LCA and LCC. This will be discussed in the following sections.

Calculating LCA

In recent years, many different LCA methods have been researched and applied (Anand & Amor, 2017). LCA can be distinguished by two methods: (1) problem-oriented approach; and

¹For the interested reader, the paper of Brander, et al. (2009) is an excellent reference since it discusses the differences between consequential and attributional LCA.

(2) damage-oriented approach. The first approach focusses on the impact of flows that are classified into environmental themes to which they contribute to, the so-called midpoints. These themes are usually related to climate change, resource depletion, and aquatic toxicity. The second approaches emphasises on the classification of a system's flow into various environmental themes, the so-called endpoints. However, the difference to problem-oriented approaches is that the damage-oriented approaches aim to model the damage of each environmental theme according to its effect on the health of the natural environment, the human health, human-made environment, and the damage of natural resources (Heijungs, et al., 2003). To conclude, the primary distinction of the two methods is that they look at different stages in the cause-effect chain, to calculate the environmental impacts of a product's system.

According to Buyle, Braet, & Audenaert (2013), three components are required in order to perform an impact assessment: (i) selection of impact categories; (ii) classification of inventory results to selected impact categories; and (iii) calculation of category indicators. Table 1 shows an example of common impact categories regarding their categorisation, classification and characterisation.

Table 1 - List of indicators from EN 15978 standard (Curran, 2006; Capaz & Seabra, 2016)

Impact Category	Inventory Flows	Characterisation	Characterisation Factor	Class
<i>Global warming</i>	CO ₂ , NO ₂ , CH ₄ , CFCs, HCFCs, CH ₃ Br	CO ₂ eq. per kg	Global warming potential	GWP
<i>Stratospheric ozone depletion</i>	CFCs, HCFCs, halons, CH ₃ Br	CFC-11 ⁵ eq. per kg	Stratospheric ozone depletion potential	ODP
<i>Photochemical oxidation</i>	NMHC	C ₂ H ₆ eq. per kg	Photochemical oxidation potential	POCP
<i>Acidification</i>	SO _x , NO _x , HCl, HF, NH ₄	SO ₂ eq. per kg	Acidification potential	AP
<i>Eutrophication</i>	PO ₄ , NO, NO ₂ , NH ₄	PO ₄ eq per kg	Eutrophication potential	EP
<i>Human toxicity</i>	Total releases to air and water	14-DCB eq.	Human toxicity potential	HTP
<i>Terrestrial ecotoxicity</i>	Toxic chemicals to rodents	14-DCB eq.	Terrestrial ecotoxicity potential	TETP
<i>Aquatic ecotoxicity</i>	Toxic chemicals to fish	14-DCB eq.	Aquatic ecotoxicity potential	AETP
<i>Abiotic depletion</i>	Quantity of fossil fuels and minerals	kg Sb eq.	Abiotic depletion potential	ADP

For the selection of environmental impact categories, in the function of the initially defined goals, the environmental impact categories can be selected, such as the depletion of abiotic resources, global warming, and human toxicity (Esnouf, et al., 2019). The second component is the classification of inventory flows to the selected impact categories, which can be done according to two methods. The first method is to divide a portion of the inventory flow to the impact categories to which they contribute. In general, this method is used in cases where effects are reliant on each other. The second method is to assign all inventory flows to all impact categories. The third component, the characterisation, targets to measure the potential impacts with regards to each selected category due to the flows that are identified in the product system. Every categorised output flow receives a characterisation factor that expresses how much that flow contributes to a specific impact category. These potential impacts are estimated at the previously mentioned mid- or endpoint levels through the characterisation factors (Capaz & Seabra, 2016). The result is expressed as an impact score in a unit common to all inventory flow within the impact category.

Following the Dutch regulations, when performing calculations for an LCA-study, the Dutch MPG should be used. The result of an MPG calculation, based on the problem-oriented LCA approach, is an environmental profile of the building consisting of eleven environmental impacts expressed in their equivalence units. Based on environmental impact categories of Table 1, two impact categories are subdivided in other impact categories: (1) aquatic ecotoxicity is subdivided into (1a) freshwater aquatic ecotoxicity and (1b) marine aquatic

ecotoxicity; and (2) the abiotic depletion is subdivided into (2a) abiotic depletion potential of composites and minerals and (2b) abiotic depletion potential of fossil fuels. Since the 1st of July 2019, the MPG has been updated in order to be in line with European determination methods EN 15804 and EN 15978, considering the inclusion of scenarios applicable to the Netherlands. The most significant change with MPG 2.0 is that the MPG 3.0 proposes a structure of the method, which is based on different modules, following the different life-cycle phases of a product. This subdivision was done in order to enable the option to make choices between products per phase. However, the specific calculation rules for calculating the environmental impacts are still being used considering the document of engineering and consultancy firm DGMR in the 2011 document '*Harmonisatie rekenregels materiaalgebonden milieuprestatie gebouwen*' and been amended by the SBK since 2013. The environmental impacts of each product component for the eleven impact categories are calculated according to the specific calculation rules. The sum of the results per product component per impact category is then added up per impact category, resulting in an environmental profile of a specific product.

The final step is to add up all the different results of the environmental profiles per impact category. This addition is calculated by using the characteristics of building products and product components. Following the MPG methodology, the form of presenting the results is done by *monetisation*. In the context of LCA studies, monetisation refers to the expression of different environmental impacts in terms of so-called shadow costs (van Gemert, 2019). Shadow costs can be identified as the theoretical valuation of costs that have to be acquired to reverse the damage caused using different materials to the environment. The different environmental impacts are expressed in equivalence units and are multiplied with the shadow cost values per environmental impact. By combining all individual shadow costs into one value, a weighted score can be merged into one indicator. A list with all the environmental impact categories with their shadow costs is shown in Table 2.

Table 2 - Environmental impact categories with their shadow costs from the MPG

Impact Category	Class	Class unit (kg eq)	Shadow cost (€/kg eq)
<i>Global warming potential</i>	GWP	CO ₂ eq.	€ 0.05
<i>Stratospheric ozone depletion potential</i>	ODP	CFC-11 ⁵ eq.	€ 30.00
<i>Photochemical oxidation potential</i>	POCP	C ₂ H ₆ eq.	€ 2.00
<i>Acidification potential</i>	AP	SO ₂ eq.	€ 4.00
<i>Eutrophication potential</i>	EP	PO ₄ eq	€ 9.00
<i>Human toxicity potential</i>	HTP	14-DCB eq.	€ 0.09
<i>Terrestrial ecotoxicity potential</i>	TETP	14-DCB eq.	€ 0.06
<i>Freshwater aquatic ecotoxicity potential</i>	FAETP	14-DCB eq.	€ 0.03
<i>Marine aquatic ecotoxicity potential</i>	MAETP	14-DCB eq.	€ 0.0001
<i>Abiotic depletion potential of composites and minerals</i>	ADPc	Sb kg eq.	€ 0.16
<i>Abiotic depletion potential of fossil fuels</i>	ADPf	Sb kg eq.	€ 0.16

There are multiple equations in order to determine the environmental impact considering the MPG-method. These different equations can be found in Appendix I. Equation 1 shows the overall environmental calculation.

$$EI^{MC} = \sum_{a=1}^i (Q_a^M \times EI_a^M)$$

In Equation 1,

EI^{MC}	Environmental impacts resulting from manufacturing and construction;
i	Number of existing materials i ;
Q_a^M	Quantity of materials used in the construction;
EI_a^M	Environmental impact of material a .

As have been mentioned before, the MPG-method makes use of so-called shadow costs. The shadow costs are calculated for each product per environmental impact category. The total shadow costs of a product for one impact category can be calculated as follows:

$$SC_{p;b} = SC_e * EI^{MC} \quad \text{Equation 2}$$

In Equation 2,

$SC_{p;b}$	Total shadow costs of a product for one impact category;
SC_e	Shadow costs, calculated per equivalence unit (see Table 2);
EI^{MC}	Environmental impacts that are resulting from manufacturing and construction.

Calculating LCC

One of the most common LCC methods used in the AEC-industry is the NPV (Langdon, 2007; ISO, 2017). The NPV can be described as the sum of the discounted future cash flows, considering costs and profits, see Equation 3. An essential variable of the NPV-method is the discount factor, which indicates the time value of money that is used to convert cash flows occurring over time (ISO, 2017). According to Langdon (2007), discounting techniques are used when the objective of the study is to compare the cost of a product over different periods in time. Furthermore, the discount rate considers inflation and possible changes in future income. This enables a considered comparison of products from different periods of time.

$$X_{NPV} = \sum (C_n \times q) = \sum_{n=1}^p \frac{C_n}{(1+d)^n} \quad \text{Equation 3}$$

In Equation 3,

C_n	the cost (i.e. net cash flow) in year n ;
q	the discount factor;
d	the expected discount rate per annum; and
p	the period of analysis

The annual net cash flow is based on the revenues, expenses, and capital costs of a building for over one year. The total annual net cash flow is the combined annual net cash flow over the entire life-cycle of a building. In general, considering the early design stages of a project, it can be stated that only rough estimations can be made to the costs after the completion of the building. The costs relating to the use, maintenance and recycling of building components corresponding to the building's life-cycle will become more evident during the development of a design. This clarification implies that applying LCC in the early design stages is difficult.

The only component of the calculation that is fixed and can be calculated in the early design stages are the acquisition costs of the building in the first year of a building's life-cycle. According to Santos et al. (2019), if the early design stages from an LCA analysis are considered within a streamlined LCC analysis (modules A1-A5 within Figure 3), then a BIM object should

only be linked to acquisition cost information. The acquisition costs can be defined as the different costs of the materials and the construction of these materials and products in a building. A way of calculating the economic impacts of the project throughout the manufacturing and construction phase can be determined according to Equation 4.

$$C^{MC} = \sum_{a=1}^i (Q_a^M \times AC_a^M) + \sum_{b=1}^j (Q_b^M \times C_b^{C2B})$$

In Equation 4,

C^{MC}	Costs resulting from manufacturing and construction phase;
i, j	Number of existing materials i, j ;
Q_a^M, Q_b^M	Quantity of materials a or b used in the construction;
AC_a^M	Acquisition cost of material a ;
C_b^{C2B}	Cost to build construction elements.

INTERPRETATION STAGE

Finally, the interpretation stage entails a systematic approach that analyses results, draws conclusions, explains limitations, and provides recommendations based on the findings of both the inventory analysis and impact assessment stages. This stage can be subdivided into three elements: (i) the identification of significant issues; (ii) the evaluation of completeness, sensitivity and consistency; and (iii) the reporting of conclusions and recommendations. By analysing the information from the inventory analysis and impact assessment phase, the identification of significant issues can be made. This information can be used in order to evaluate the completeness, sensitivity and consistency of the LCA-study. The final step is to draw conclusions and make recommendations based on the results found in the study.

There are numerous techniques available that can be utilised to support the decision-making process. An often-used technique is the *decision tree technique*. With decision trees, a graphical presentation is given, which illustrates hierarchal interrelations. Buchert et al. (2015) mention that the applicability of decision trees with regards to life-cycle analysis is useful since it can be used to model process alternatives for every phase in the product lifecycle. Each time a design parameter is set, the different alternatives are limited, steering to other decision pathways. Figure 6 shows an example of a decision tree, based upon the three indicators of life-cycle sustainability: economic, social and environmental.

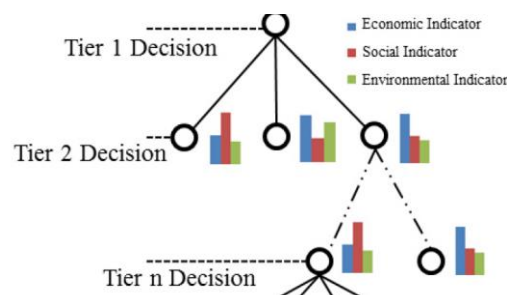


Figure 6 - A decision tree based on the three pillars of LCSA (adopted from Buchert et al. (2015))

Decision trees are mainly based upon not only one criterion but on multiple criteria. Just by looking at the environmental factors alone, it can be stated that there is a multitude of contributing factors that make up an environmental system. Network theory, also known as *graph theory*, can be described as a way of a graphical expression in which to demonstrate

complex systems. For example, Baez et al. (2018) classify network theory as the study of complex interacting systems that are represented by graphs, interlinked within a structure. The definition of a graph is that it contains a few vertices that are connected by edges – see Figure 7. Through the representation of a problem within a graph, the problem is approached from multiple perspectives which optimally leads to the simplification of the problem. It can be stated that network theory can be used as complementation for multiple criteria decision-making since it aids in a graphical output for the complex problems that are analysed.

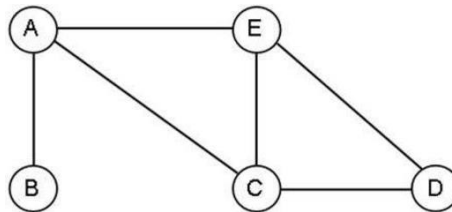


Figure 7 - A network graph based on the different vectors and edges

Kim and Holme (2015) claim that network theory can be used as useful complementation to LCA analysis for complex allocation, which implies that for network theory for LCA-analysis of a design is only useful during advanced stages of design. Besides, Heijungs (2015) strengthens the theory of Kim and Holme (2015) by mentioning that it is challenging to include the environmental aspect using the network theory. Furthermore, Heijungs (2015) mentions that *“it is not entirely clear what the resulting network indicators imply for specific LCA cases”*.

As have been mentioned earlier, another technique that can aid the multiple criteria required by environmental and economic studies is called multiple criteria decision-making (MCDM). MCDM supports a framework that is utilised to assess multiple criteria. It is useful for presenting the relationship among different alternatives, relating it to the general objective(s). MCDM aids in the assessment of various alternatives and determines which is the most optimal solution. However, it must be stated that in general, one optimal solution based upon different objectives is not possible. Mainly, the decisionmaker must rely on their judgement and preference in order to differentiate between solutions. Therefore, MCDM is usually classified into two main categories; (1) multi-attribute decision-making (MADM), and (2) multi-objective decision-making (MODM). With regards to the MADM, problems consist of a finite number of alternatives, explicitly known at the beginning of the solution process. These alternatives are represented related to their performance in multiple criteria. The problem can be described as acquiring the best alternative for a decision-maker or finding a set of reasonable alternatives. Within the AEC-industry, a popular MCDM method is the Analytic Hierarchy Process (AHP), which analyses problems according to a specific comparison scale. This scale helps decision-makers to find the best alternative suited to their objectives and preferences and does not prescribe one ‘correct’ alternative.

On the other hand, within MODM, the alternatives are not explicitly known. An alternative can be found by solving a mathematical model, in which the number of alternatives is either infinite or typically extensive. If all the objective functions are linear in terms of decision variables, this variation leads to multiple-objective linear programming (MOLP). With regards to implementing linear programming into LCA methods, Azapagic and Clift (1999) proclaim that *“linear programming can be successfully combined with LCA to provide a powerful tool for analysing and managing environmental performance of a system while still satisfying other requirements, such as economic performance”*. Furthermore, Budzinski, Sisca and Thrän

(2019) mention that the benefit of using LP in a combination of LCA is to provide a broader and systematic assessment of the consequences. Besides, Budzinski et al. (2019) mention that combining LCA and LCC targets within a multi-objective programming approach enables the user to address and assess environmental and economic trade-offs in decision-making. By utilising MOLP with regards to environmental and economic targets, the decision-makers can be supported to optimise their designs in early design stages.

2.3 BUILDING INFORMATION MODELLING

The concept of Building Information Models (BIM) has received much attention in recent years. Although BIM has made its appearance in the 1970s, BIM has initially been developed in 2000s in order to visualise the built environment (Jrade & Jalaei, 2013). The first publication, which provided a concept that we now know as BIM, was the description of a working prototype called the *“Building Description System”* by Charles M. Eastman (1976). Within this prototype, elements were interactively defined and related, which had as a result that all changes had to be made only once in order to make changes in all other drawings.

Furthermore, data about costs, quantities and materials could easily be generated as well. This first publication was the founder of the contemporary concept of BIM. According to Gkatzios (2018), BIM marks the shift from line-based geometry, which is represented by Computer-Aided Design (CAD) tools, to an object-based geometry. This shift stimulated the accessibility and operability of semantic data into the visualisation of a model with all its characteristics (Jrade & Jalaei, 2013). Through the years, BIM has been defined in various ways in the literature. At this time, BIM is extensively implemented within the AEC-industry for a large variety of processes. Although differently interpreted by each person, Koppula (2013) claims that people consider BIM as nD-modelling, collaboration, virtual reality, clash detection and control, coordination, and information management. According to Ghaffarian Hoseini et al. (2017), nD-modelling related to BIM consists of 3D visualisation, 4D construction sequencing, 5D cost estimating, 6D sustainability assessment and 7D facility management. BIM implementation has been considered as an effective way of working collaboratively throughout the whole lifecycle of the building.

2.3.1 THE CONCEPTUAL BASE OF BIM

BIM is considered to have enhanced the efficiency in the data generation of a building's lifecycle. In the early design stages, BIM has allowed users to share detailed information and knowledge of the building (Azhar, Hein, & Sketo, 2008; Ghaffarian Hoseini, et al., 2017). Furthermore, BIM has supported engineers in the decision-making of designing an integrated project at an optimum level (Gkatzios, 2018). Moreover, a BIM model includes necessary semantic information about the project itself. Therefore, decision-making can be encouraged by smart design feedback and solutions so that each user optimises his performance. Ghaffarian Hoseini et al. (2017) states that although BIM has almost been completely implemented in the AEC-industry, its potentials have not been fully exploited yet. AEC-related companies include the requirements in significant investments by in software, hardware and training (Weisheng, 2017). Moreover, the problem of low usage is further aggravated by technical reasons. Two of the main problems of implementing BIM are interoperability difficulties and the lack of software usability (Grilo & Jardim-Goncalves, 2010; Bryde, Broquetas, & Volm, 2013; Ciribini, Ventura, & Paneroni, 2016). A need to standardise the BIM process and to define the guidelines for its implementation is critical. The building model usually goes through a step-wise process in which the model contains varying levels of detail

based on the stage of the project development (Solihin, Eastman, & Lee, 2016). The *Level of Development* (LOD) specification is a reference that facilitates practitioners in the AEC-industry to specify and communicate with a certain level of clarity, the content and reliability of BIM-models at several stages in the design and construction process (BIMForum, 2019). An element's geometry and information content are generally following the five levels: (i) LOD 100 – Conceptual; (ii) LOD 200 - Approximate geometry; (iii) LOD 300 - Precise geometry; (iv) LOD 400 – Fabrication; and (v) LOD 500 - As-built and facility management (Reddy, 2012). LOD identifies how much information is known about a model element at a given time.

2.3.2 THE ROLE OF BIM IN THE AEC-INDUSTRY

A BIM-model combines information which is relevant to the building, considering its functional characteristics, its graphical features, its physical features, and project life-cycle information, in a series of *smart objects*. Smart objects can be described as objects that contain their functions in their property set in order to interact with users and other objects (de Vries, Allameh, & Heidari Jozam, 2012). In general, BIM-models are realised by using object-oriented software such as *Revit*, *Sketchup*, or *ArchiCAD* and consist of parametric objects that represent the different building elements (Ghaffarian Hoseini, et al., 2017). The uniqueness of objects can vary in geometric and non-geometric attributes regarding functional, semantic or topologic information (Azhar, Nadeem, Mok, & Leung, 2008). Models developed by different design disciplines can be connected by combining their models. This combination of models can be considered an essential part of BIM; all the relevant information about a building project is presented in one central model or data repository, which can be used by all stakeholders during the entire lifecycle of a building (Paavola & Miettinen, 2018). Due to the promoting role of BIM towards collaboration and information sharing among project participants, the production efficiency of design and the construction processes can be improved (Eastman, Teicholz, Sacks, Liston, & K., 2011; Barlish & Sullivan, 2012). Besides, a parametric BIM model facilitates the production of accurate results. When designers encounter intricate geometric designs, instant feedback to these changing variables can be given more easily. The digitised and parameterised characteristics of BIM positively affect in terms of that project designers are enabled to thoroughly analyse the influences of the environment and energy, for example (Chien, Wu, & Huang, 2014).

By looking at the development of implementing BIM in the AEC-industry, it can be concluded that BIM is rapidly growing through government initiatives to promote and obligate the use of BIM. For example, the United States General Services Administration formulated the National 3D-4D-BIM Program which established policies to mandate BIM adoption for all Public Buildings Service projects (Yee, Matta, Kam, Hagan, & Valdimarsson, 2009). The Government of the United Kingdom has restricted all procurement project in the public sector must implement a BIM approach (Akinade, et al., 2017). The Danish government invests heavily in the research and development of BIM. Furthermore, all the Danish government clients are all required to adopt BIM in their projects (Smith, 2014). With regards to the Dutch government, the Rijksvastgoedbedrijf (RVB) states that a standardised RVB BIM-norm should be used for Design, Build, Finance, Maintain and Operate (DBFMO) and Design, Build and Maintain (DBM) contracts in the AEC-industry (Rijksvastgoedbedrijf, 2019). As DBM is quickly becoming standardised on governmental as industrial level, any promising innovation must be BIM compliant (Ajayi, et al., 2014).

2.3.3 INDUSTRY FOUNDATION CLASSES

BIM-models include a vast amount of information and serve as a technical basis for the participants in a BIM process, such as engineers, decision-makers or operators. The data is used throughout the entire life-cycle of a building (Eastman, Teicholz, Sacks, Liston, & K., 2011). For interoperability purposes, the building information must be accessible in a standardised and open way, which can be ensured through Industry Foundation Classes (IFC). IFC standard is a neutral and open data exchange standard of objects which facilitates the exchange of building-related data among various platforms used in the AEC-industry.

IFC was developed by BuildingSMART international and is a widely adopted format for the exchange of building models. The specification of IFC has been defined by ISO Standard for the Exchange of Product model data (STEP) exchange format and uses the EXPRESS data definition language for the definition of data semantics. This specification allows a taxonomical classification among the different classes, entities, properties and attributes of a model. The IFC data model is fundamentally constructed in a hierarchical structure (Theiler, Bargstädt, & Smarsly, 2018). However, its object-oriented approach allows complex relationships to exist between different entities (Dimyadi, Spearpoint, & Amor, 2008). Entities may be physical objects such as walls, floors, and windows, or more conceptual entities such as spaces or processes.

In addition to these objects, the IFC schema provides a predefined set of types, rules, and functions in order to capture relevant information to the domain of interest, the so-called *inheritance*. The inheritance determines that a specific class is a subtype of another class, which consists not only its attributes and relationships, but also inherits those from its predecessors (Ismail, Nahar, & Scherer, 2017). In other words, the functionality of the inheritance is that it can either provide a relationship between different objects or that it contains value-driven data types, which contain information such as areas and widths of building elements, or the categorical names of certain building elements. Figure 8 shows a part of the IFC model structure, displaying main building elements expressed in an EXPRESS structure. With regards to the different identifiable objects in the IFC schema, the so-called *IfcRoot* class forms the predecessor. According to Ismael, Nahar, & Scherer (2017), three fundamental classes derive from this *IfcRoot* class: (i) *IfcRelationship*; (ii) *IfcPropertyDefinition*; and (iii) *IfcObjectDefinition* (Ismail, Nahar, & Scherer, 2017). These three classes shape the first layer of the *IfcRoot* hierarchy. *IfcObjectDefinition* defines any semantically given object or process, while the *IfcPropertyDefinition* covers all characteristics that are being assigned to objects. *IfcRelationship* is accountable for the connectivity among objects.

In order to provide a linkage between BIM-data on the one hand and LCA and LCC on the other hand, the most essential component to extract from the IFC-model is the *Bill of Quantities* (BoQ) and the identification of the building elements (Soust-Verdaguer, Llatas, & García-Martínez, 2017). The BoQ provides different quantities within the IFC-model, which can be used for the calculation of the environmental and economic impacts. By linking the BoQs to the related identification of the building elements, specific calculations can be made. However, the quantities in an IFC-schema are not directly attached to specific objects. These quantities can be found in different property sets, such as the *IFCElementQuantity* and the *IFCPropertySet*. The most efficient way of deriving quantities would be to extract the quantities from the *IFCElementQuantity* property set. This property set defines a set of derived

measures of the physical characteristics of an object. This quantification depends on the software package that is used to export the IFC-model. With regards to the identification of the building elements, the classification of the objects is mainly provided through the *IFCClassification*. The relationship used for classification is *IFCRelAssociatesClassification*.

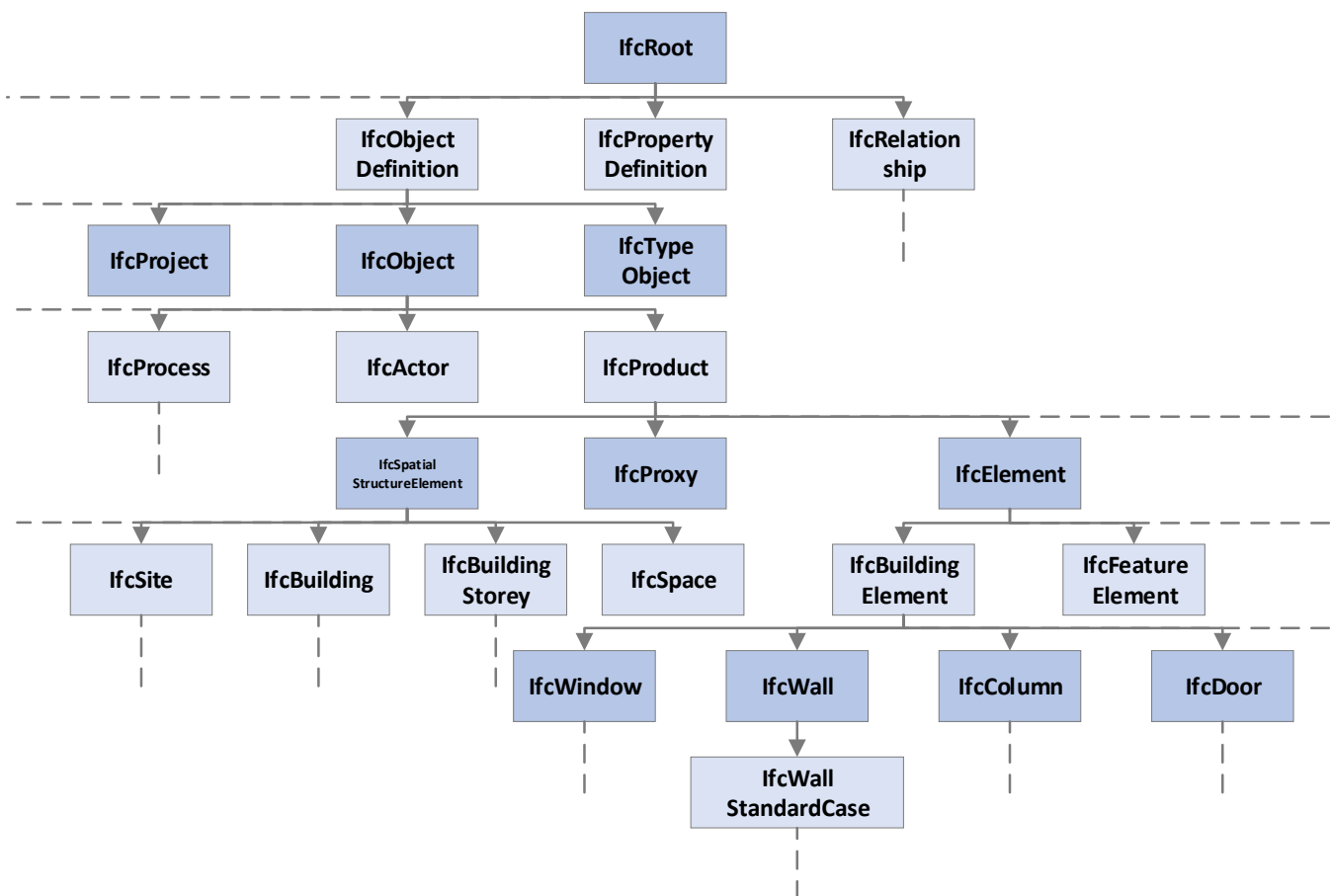


Figure 8 - EXPRESS diagram showing a part of the IFC-hierarchy (adapted from Dimyadi et al. (2008))

2.4 THE INTEGRATION OF LCA, LCC AND BIM

Since both LCA and LCC approaches are based upon life-cycle thinking, assuming a specific lifetime for materials and the building, the approaches can be considered appropriate to be combined. This appropriateness of combination is because the LCA and LCC approaches can provide knowledge of the potential life-cycle costs and environmental impacts for alternative designs simultaneously. The integration of LCA and LCC may, for instance, be used to make alternative decisions within a design, to identify technical solutions that meet an environmental target to the least cost and to evaluate a building investment (ENSLIC, 2010).

LCA and LCC have predominantly been used by individual assessment tools, but there has been a number of academic researches and commercial tools that started to link both forms of assessment in order to execute their analysis of the building performance (Schmidt, 2018). According to Langdon (2007), LCA and LCC can either be used in conjunction with each other on a broader evaluation process or be utilised to process and implement each other's input. Ahmed and Tsavdaridis (2018) mention that in order to integrate the overall structure of an environmental-economic performance assessment, the environmental and economic performance needs to be calculated individually, as can be seen in Figure 9. This figure shows an overview of the content of a complete environmental-economic performance assessment

of an LCA and LCC-study. However, the application of LCA and LCC still needs to be improved and simplified with regards to the different design stages of a building, especially during the early design stages (Soust-Verdaguer, Llatas, & García-Martínez, 2016; Santos, Costa, Silvestre, & Pyl, 2019). This development needs to happen in order to stimulate the implementation of LCA and LCC in the AEC-industry.

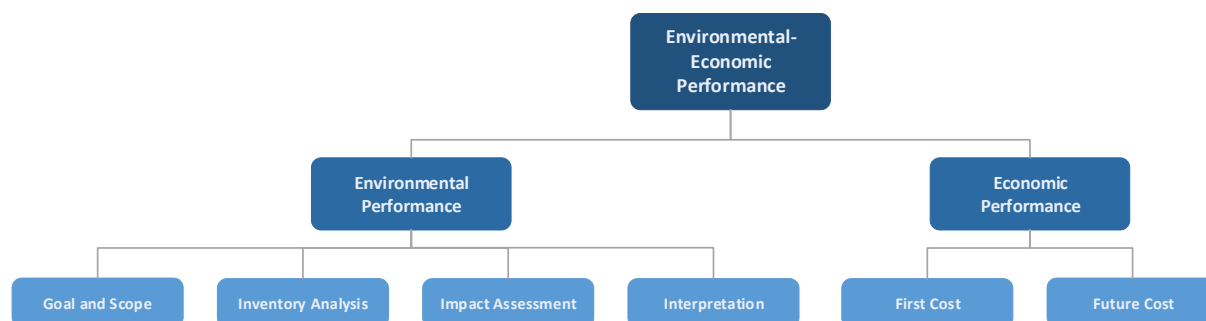


Figure 9 - Environmental-Economic Performance Assessment (Ahmed & Tsavdaridis, 2018)

2.4.1 STUDIES THAT USE LCA AND LCC INDEPENDENTLY

According to Schmidt (2018), studies that include LCA and LCC as part of a building performance assessment often use the terminology, datasets and calculation methods inherent to both approaches. These calculations and assessment are often performed independently from each other as a fundamental part of the assessment. After this, the results of the LCA and LCC are then compared and evaluated, based upon the analysis of the relationship between LCA and LCC and how these two entities can be evaluated simultaneously in order to support the decision-making process (Fouche & Crawford, 2017). Mithraratne and Vale (2004) performed a study in order to include detailed embodied inputs supported by an additional tool to aid assessment of initial embodied energy and cost of building materials along with the operating energy and cost. The study found that embodied energy was significantly higher in an energy-efficient house. On the other hand, the operating costs were reduced simultaneously, which resulted in a decrease in the overall energy demand. However, a drawback of this study is that the assumptions made within the tool were unclear, implying that the results can be considered questionable.

Ristimäki et al. (2013) proposed an LCA and LCC framework that initially can be used separately, integrating the results afterwards in order to provide an assessment based on both environmental- as well as economic performance. At first, LCC data was collected and calculated manually considering the *net present value* (NPV) methodology over 25 years, 50 years and 100 years. Besides, the life-cycle affordability of each option was calculated by including replacement, maintenance, energy and investment cost. In addition to the LCC assessment, a streamlined LCA assessment was performed, calculating the different greenhouse gas emissions for various construction activities. The combined results of both the LCC and LCA were then presented in one graph, showing the net present cost on the left and the greenhouse gas emissions on the right, which is illustrated in Figure 10. Ristimäki et al. (2013) concluded that by selecting slightly higher investment costs provides a significant decrease in energy costs and emissions. Besides, they reckoned that the economic and environmental life-cycle effects followed similar developments and are closely connected. However, a disadvantage of this study is the calculations, since the use of different software packages uses their datasets and internal calculations, which are often unclear to the user. Again, this means that the results were unclear, which implies a lack of interpretation.

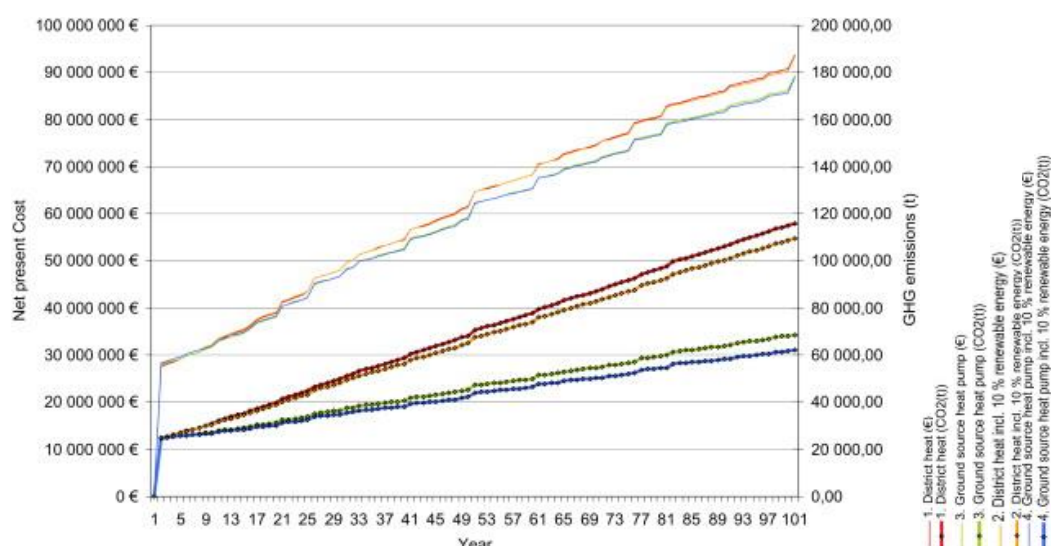


Figure 10 - Cumulative LCC and LCA outcome over 100 years (Ristimäki et al. (2013))

Schwartz et al. (2016) completed a residential study in the UK in which the analysis was made in two parts. First, they examined the embodied carbon emissions and costs before they examined the operational carbon emissions and cost in order to make a distinction in the life-cycle carbon footprint and the LCC. Once the values of both the life-cycle carbon footprint as the LCC were determined, the relationship between the different values was evaluated. By using multi-objective algorithms, different building strategies were evaluated from both the economic as well as the carbon perspective. The overall conclusion of the study was that even though a selected strategy may save carbon or cost, the strategy also must consider the user needs and preferences in order to determine the most optimal solution.

2.4.2 STUDIES THAT INTEGRATE LCA AND LCC

Besides independently executing LCA and LCC studies and evaluating and combining the studies later, frameworks in which LCA and LCC are linked with each other are gaining interest by researchers. For example, Heijungs et al. (2013) provided an explicit and transparent description of how to calculate the life-cycle cost and the value-added across the life-cycle. The proposed matrix-based approach tried to combine both physical as well as monetary flows for adapting computational structures of environmental LCC as well as LCA. However, Heijungs (2013) concluded that the integration of the two methods is complicated due to the differences in purpose, terminology, framework and calculation rules, for example.

An Italian study created a tool for the systematic sustainability assessment based on LCA and LCC (Petrillo, et al., 2016). Petrillo et al. (2016) performed their analysis in three different phases; (1) Analytical phase, (2) Modelling phase, and (3) Assessment phase. The first phase provided the initial analysis and characterisation, which included scenario definition and choice of indicators, corresponding to the first two stages of an LCA and LCC-study (see section 2.2.2). The second phase carried out an LCA and LCC-study, resulting in an extensive collection of data. The third and final phase utilised some methods in order to integrate the results from the previous phase, to establish a multi-criteria analysis and flexibility analysis, and to conclude the different options within a sustainability classification. Although this study provides a clear framework on how to integrate LCA with LCC, it did not provide a clear explanation of deciding the potential indicators for environmental and economic analysis in phase two and weighting and scaling factors used for the sustainability rating in phase three.

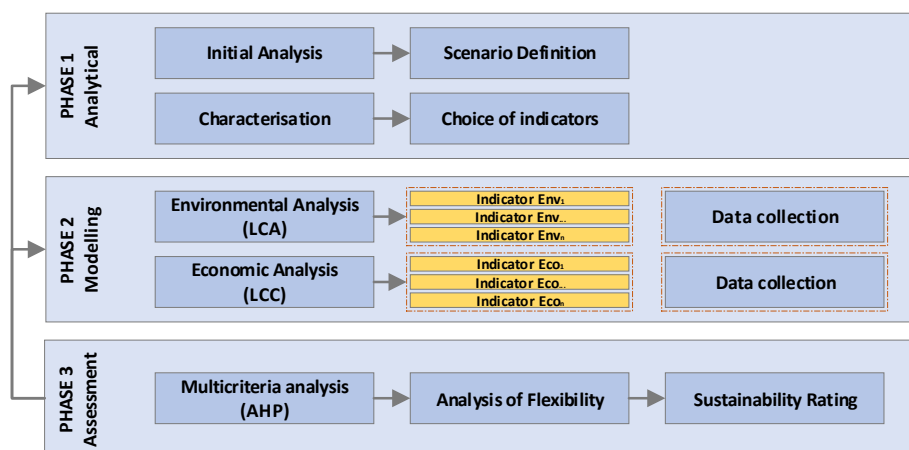


Figure 11 - Framework for integrating LCA and LCC (adapted from Petrillo et al. (2016))

Another Italian study executed by Savino et al. (2017), aimed to create a new model for the environmental and economic evaluation of renewable energy systems. This model is based on a colour-coded matrix, combining the concurrent environmental and economic viability of renewable energy systems. This matrix is divided into four regions, varying from the most optimal region to the intolerable region, which is shown in Figure 12. Different renewable energy systems were analysed and presented in the matrix, generating LCA and LCC results which provide the user to make considered choices based upon different options. However, this study fails to incorporate parameters based on user preferences. By including parameters based on user preferences, the results can be filtered and directed towards specific solutions.

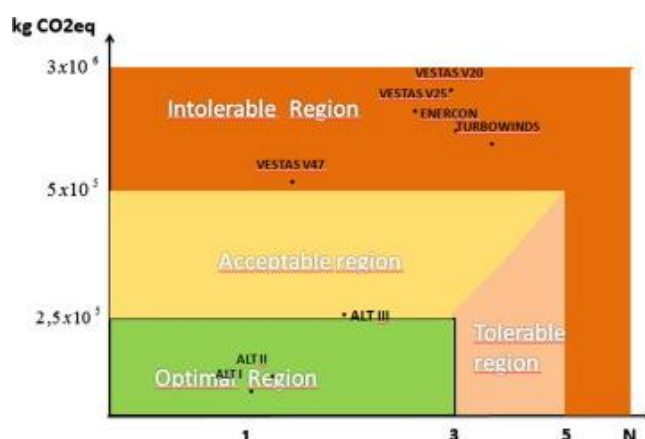


Figure 12 - Environmental and Economic Impact Assessment matrix (Savino et al. (2017))

Research by Santos et al. (2019) explored the potential of BIM as a repository for integrating LCA and LCC. A BIM-based LCA and LCC framework was proposed, which was used in order to enable automatic calculations for environmental and economic assessment within a BIM-environment. Principally, this research has examined the potential of parametric modelling, in which LCA and LCC data is incorporated within the BIM-model. Although the framework proposes an interesting way of integrating LCA with LCC through the use of BIM, Santos et al. (2019) conclude that *"BIM-LCA and LCC tools require more information than is usually contained in BIM models to properly conduct an LCA and/or LCC analysis within a BIM-based environment."* In other words, the integration of LCA and/or LCC analysis within a BIM-based environment is hard to perform when the LOD of building elements is low. In order to make a comprehensive LCA and LCC analysis within a BIM-based environment, the building elements require to have more detail on material levels, such as the mechanical aspects of materials, quantity aspects, and environmental impacts from the manufacturing.

Furthermore, within the framework of Santos et al. (2019), in order to show the distinction of LCA and LCC studies, two different LCA and LCC studies were performed; the streamlined and complete analysis. With regards to these studies, a complete study can only be performed when an LOD of a BIM-model is 300 or higher. For the streamlined analysis, generic LCA and LCC data can be used to make environmental and economic calculations which enables the LCA and LCC study within the early design stages.

2.4.3 STUDIES THAT INTEGRATE LCA AND LCC WITH BIM

A variety of studies present complex and data-intensive frameworks and methods, such as Heijungs et al. (2013) and Petrillo et al. (2016), which are not suitable for implementation in early design stages. The implementation of life-cycle environmental and economic optimisation during early design stages is key as the potential for change decreases rapidly throughout the building life-cycle (Schmidt, 2018).

One way of overcoming this issue is by integrating BIM with LCA and LCC. The ideal workflow for LCA and LCC integration must provide an automatic interaction between a BIM model and the data about environmental, economic impacts (Llatas, et al., 2019). Röck et al. (2018) proposed a link between LCA with BIM in order to calculate embodied environmental impacts. This link was recommended to be a product of the total area of a building, obtained from a BIM model, and the environmental impact values which were extracted from an LCA database. Soust-Verdaguer et al. (2018) set up a BIM-based LCA method which made a comparison between different environmental performance alternatives of the building envelope during their life-cycles. The proposed workflow by Soust-Verdaguer et al. (2018) was based upon the integration of the BoQ extracted from the BIM model with environmental data which was used to execute an environmental impact calculation. Seo, Tucker and Newton (2007) developed a system, LCADesign, which takes off quantities of building components from an object-oriented 3D-model, calculates inventory of the components of the building products, and links the results to an LCI database for environmental impact assessments.

Furthermore, this system aids in the visualisation of environmental impacts of the building, resulting from the combination of choice of materials design that is used in construction. Jade and Jalaei (2013) proposed a framework that integrates BIM and LCA application with an inventory database for designing a sustainable building, in which the output formed an environmental impact. Bueno et al. (2018) developed an application that linked Dynamo that enables designers to create visual logic to explore the potentials of parametric conceptual designs, with Microsoft Excel to explore the different environmental contours for decision-making in the early stages of design. To conclude, researches show the scarce existence of methods that integrate LCA and LCC studies with BIM, which can be seen in Table 3.

Overall, different case studies analysis agree that the more complex data structure and information is, the more difficult it is to automatise the integration of LCA and LCC with a BIM (Llatas, et al., 2019). Although, many research papers emphasise the importance of integrating LCA and LCC in a BIM environment to facilitate more efficient decision-making (Basbagill, Flager, Lepech, & Fischer, 2013; Antón & Díaz, 2014; Anand & Amor, 2017; Nwodo, Anumba, & Asadi, 2017; Santos, Costa, Silvestre, & Pyl, 2019). Soust-Verdaguer et al. (2017) proclaim that *“one of the most important uses of BIM models in the LCA application is to obtain the bill of material quantities”*. Moreover, BIM can have a supporting function in the early design

stages, as it facilitates the use of LCA in the AEC-industry. During these early design stages, the potential for examining various alternatives, implementing changes and improving environmental and economic performances is the highest (Anand & Amor, 2017). With regards to LCA, BIM can give timely information about whole life environmental issues as the design changes and develops. It can also support in the decision-making process at the beginning periods of the plan (Azhar & Brown, 2009; Anand & Amor, 2017). Because BIM takes multi-disciplinary information to be covered by one model into consideration, it provides an opportunity for environmental and economic assessment tools, such as the LCA and LCC, to be incorporated in the design process (Jrade & Jalaei, 2013; Nwodo, Anumba, & Asadi, 2017).

Table 3 - List of LCA and LCC studies integrated with BIM (adapted from Llatas et al. (2019))

Reference	Design stage	Dimension of Sustainability
<i>Basbagill et al. (2013)</i>	Early design stage	Environmental
<i>Bueno et al. (2018)</i>	Early design stage	Environmental
<i>Jrade et al. (2013)</i>	Early design stage	Environmental
<i>Röck et al. (2018)</i>	Early design stage	Environmental
<i>Santos et al. (2019)</i>	Early design stage and Detailed stage	Environmental and Economic
<i>Seo et al. (2007)</i>	Early design stage and Detailed stage	Environmental
<i>Shin et al. (2015)</i>	Detailed stage	Environmental and Economic
<i>Soust-Verdaguer et al. (2018)</i>	Detailed stage	Environmental

2.4.4 INTEGRATION APPROACHES FOR INTEGRATING BIM WITH LCA AND LCC

In the literature on BIM integration with LCA and LCC, three main approaches can be mentioned (Santos, Costa, Silvestre, & Pyl, 2019). The first approach is the utilisation of a variety of applications to conduct the LCA and LCC analysis individually (Zuo, et al., 2017; Fauzi, Lavoie, Sorelli, Heidari, & Amor, 2019). This approach is labour-intensive and takes more expertise in a variety of applications. Therefore, this approach is not recommended in order to make a well-considered analysis between LCA and LCC.

A more streamlined approach is the second approach, in which the integration of LCA and LCC within a BIM-based environment utilises a data exchange standard that comprises the BoQ and other properties of BIM model elements, which are then processed in an external application or database (Antón & Díaz, 2014; Bueno, Pereira, & Fabricio, 2018; van Gemert, 2019; Santos, Costa, Silvestre, & Pyl, 2019). This BoQ provides project-specific measured quantities of the items of a design, which can be measured in number, length, area, volume, weight or time, for example. In addition, by only using the BoQ from a BIM-model, BIM-based LCA and LCC applications related to this second approach only need to have the right classification of data in order to link the data with LCA and LCC databases. Thus, the import of all the BIM-models within an LCA and LCC application must be done considering a corresponding classification. A drawback of this second integration approach, mentioned by the Santos et al. (2019) was that *“BIM-LCA and LCC tools require more information than is usually contained in BIM models to properly conduct an LCA and/or LCC analysis within a BIM-based environment.”* This higher level of information means that, in order to make a more accurate analysis, the BoQ should contain all the detailed information present in a design (Santos, Costa, Silvestre, & Pyl, 2019). In other words, although a calculation can still be made based on generic data, it is recommended to have a BoQ with a higher LOD to make more accurate calculations. Besides, incorporating valuable data within a BIM-model means that the BoQ can only be exported from a single BIM software package. By using only one BIM software package, the interoperability between the use of different BIM-models from

different software packages can be a challenge. This challenge can, however, easily be solved by linking external databases that are corresponding to the BIM-model. Therefore, linking different components with each other through different platforms can help to manage and to calculate the environmental and economic impacts in a structured way. Besides, with regards to decision-making, an external database with environmental and economic data is necessary in order to make a well-considered comparison with the data from the BIM-model.

A third approach, in the literature mostly linked to LCA, suggests the inclusion of LCA information within BIM which could be used to represent an initial step towards environmental integration in BIM (Díaz & Antón, 2014). As noted by Santos et al. (2019), *“the lack of information within BIM models not only leads to ineffective decision-making but also prevents automatic simulations of the project in several domains”*. Instead of utilising methodologies where BIM-models are just used for their geometric- and material data, efforts must be made to incorporate valuable data in a model (Antón & Díaz, 2014; Santos, Costa, Silvestre, & Pyl, 2019). Santos et al. (2019) conclude that if a BIM model also comprises information regarding the environmental and economic impact of materials, automatic streamlined analysis of the whole project would at least be possible. With regards to a complete study, however, it would be required to insert specific information of the project manually. Such information must be put in a structured way, consisting of the aspects of the system's boundary, type of LCA and LCC data, and the environmental and economic impact assessment method used. However, an IFC schema does not contain suitable properties to store the type of information required by each environmental impact category yet. According to Santos et al. (2019), a minimum of 137 IFC-properties is required for including LCA and LCC data, where at least 128 properties have to be created and included in the IFC schema. Therefore, in order to implement a third integration approach, additional property types that contain the different LCA and LCC data have to be created.

2.4.5 THE IPO-MODEL

At the highest level of abstraction, the different approaches of BIM-LCA and LCC integration all comprise of three interlinked components (Negele, Fricke, Schrepfer, & Härtlein, 1999). The three interlinked components comprise: (i) two input sources, being the BoQ from a BIM model, and a combination of an external database consisting of both environmental and economic data on building products or materials; (ii) an calculation assessment module which calculates the environmental impacts based on the linkages between the two data inputs; and (iii) a dashboard that presents the results. The so-called *input-process-output* model or IPO-model is a widely used approach, mainly for systems analysis and software engineering, which describes the structure of an information process (Negele, Fricke, Schrepfer, & Härtlein, 1999).

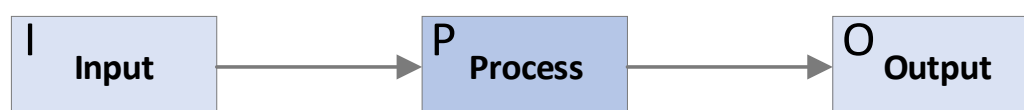


Figure 13 - Input-Processing-Output method (adapted from Negele et al. (1999))

The input stage comprises all the data flow that is required to generate a specific output. With regards to the input of BIM data towards LCA and LCC, the two most essential components are that of (i) the accessibility of the BoQ; and (ii) the identification of building elements (Soust-Verdaguer, Llatas, & García-Martínez, 2017). The accessibility of BoQ provides the opportunity for scaling the environmental impacts based on the quantities. These quantities

are based upon building elements which are identified and linked to specific products or materials in an external database.

The process stage can be interpreted as the different activities and operations that are required to process the different input variables, such as calculations and decision-making. The output will comprise the results of the different activities and operations made and proposed in the process stage. This output stage mainly differs in feedback (Santos, Costa, Silvestre, & Pyl, 2019). In the current practice, BIM-LCA and LCC applications provide feedback varying from results of calculations, to the visual representations of design changes (Nwodo, Anumba, & Asadi, 2017). However, the feedback mainly lacks in terms of potential changes to a design. By providing suggestions on how to make design changes, practitioners are enabled to make well-considered decisions.

2.5 FEEDBACK APPROACHES

Adapted from Dempsey, Driscoll, and Swindell (1993), the complexity of feedback plays a role in providing designers knowledge on how to use the feedback. Based on the complexity of feedback, Dempsey et al. (1993) distinguished five different feedback complexity approaches. These complexity approaches are: (1) no feedback; (2) simple verification feedback or knowledge of results; (3) correct response feedback; (4) elaborated feedback; and (5) try-again feedback. The *no-feedback* approach means that the designer is presented a question about the design and is required to respond, but no suggestion is provided as to the correctness and completeness of the designer's response. With regards to the *simple verification feedback*, the designer is informed about the correctness of the response based upon the knowledge of results. An exciting approach of feedback is the *correct response feedback*, which informs the designer what the correct response on the design should be. By providing such an approach, the designer is enabled to make decisions based upon the system's database. The *elaborated feedback* approach goes even further because it explains why the designer's response is correct or incorrect, and it allows the designer to review their design. Lastly, the *try-again feedback* approach informs the designer when an incorrect response on the design is given and allows the designer to make one or more additional attempts on adapting the design.

It can be stated that many existing applications are using a simple verification feedback approach, in which the designers are provided with the results of the environmental or economic impact of their design. However, designers are not given any direction in making a well-considered sustainable decision. On the other hand, some applications in the AEC-industry that have integrated a BIM-LCA and LCC approach are using the try-again feedback approach. A feedback-providing application is *SimaPro*. *SimaPro* allows to collect, analyse and monitor the sustainability performance of products (Ciroth & Franze, 2009). According to Starosta-Patyk (2015), *SimaPro* provides decision-makers with the possibility to "*analyse complex life-cycles in a systematic and transparent way*", and to "*measure the environmental impact of products and services across all life-cycle stages*". However, although its extensiveness of information, *SimaPro* cannot implement a BIM-model within the application.

CAALA for example, a German plugin within certain BIM-software that provides graphical output for energy- and environmental effects, calculates both the energy requirements for the operation of a building and the production, replacement and removal of the building materials during the early design phases. However, it must be stated that the output of this plugin only provides real-time calculations and does not provide any guidance on what type

of design changes must be made whatsoever; it gives the designer the option to adapt the design repeatedly but not the reason why. Furthermore, CAALA is only applicable to the BIM-software *SketchUp* and *Rhinoceros*, which limits these practitioners of using other software packages. By using IFC-models, the exchange of (specific) BIM information between various software applications of different participants of a design process can be established.

2.6 CHALLENGES OF INTEGRATION

It is evident that in the past decades LCA and LCC have been developed quite independently, and so there are variations in terminology, methodology, and calculation rules (Heijungs, Settanni, & Guinée, 2013). Therefore, if LCA and LCC studies are performed separately towards decision-making, discrepancies can only emerge within either the environmental aspects of LCA studies or the economic perspectives of LCC studies (Settanni, 2008). However, when LCA and LCC studies are conducted together considering the execution by the same practitioners, in the same software-packages, and within the same databases, inconsistencies of decision-making based on both studies may provide a barrier in terms of efficacy, reproducibility, and comprehensibility (Heijungs, Settanni, & Guinée, 2013; Santos, Costa, Silvestre, & Pyl, 2019).

The challenge of combining LCA and LCC is to overcome the differences in, for example, the identification and categorisation of the principles and the system boundaries, and the alignment of LCA with LCC. Some literature has addressed the challenge and attempted to harmonise the set-up and principles of LCA in combination with LCC (Hunkeler, Lichtenvort, & Rebitzer, 2008; Swarr, et al., 2011). For example, Hunkeler et al. (2008) described LCC as the assessment tool for costs that correspond with the entire life-cycle of a product that is directly encompassed by different actors in the life-cycle. This description is in line with the definition of ISO, which defined LCA as a collection and evaluation of different inputs, outputs and potential environmental impacts of a product system throughout its whole life-cycle (ISO, 2006). Table 4 presents different challenges identified by various researches of integrating BIM, LCA and LCC.

Table 4 - Challenges of integrating LCA, LCC and BIM

Issues	LCA LCC BIM	Azapagic (1996)	Díaz et al. (2014)	Guinée et al. (2016)	Häkkinen et al. (2008)	Iacovidou. et al. (2018)	Llatas et al. (2019)	Krigsvoll (2007)	Norris (2001(b))	Nwodo et al. (2017)	Santos et al. (2019)	Schmidt (2018)	Soust-Verdaguer et al. (2017)	Rahmani Asl et al. (2015)	Röck et al. (2019)
Lack of integration methods	LCA LCC BIM	•					•	•	•	•	•				•
A high amount of data requirements	LCA LCC BIM				•				•						
Loss of information	BIM						•				•	•		•	
Lack of transparent inputs and calculations	LCA and LCC			•		•					•	•			
Lack of visually integrated LCA and LCC results	LCA and LCC									•		•	•		
Lack of early design stage applications	LCA LCC BIM		•	•	•					•	•	•	•	•	•
Lack of correct response feedback	LCA and LCC	•				•			•	•		•			
Lack of decision-making based on user preferences	LCA and LCC	•	•	•				•	•	•	•	•	•		•

2.6.1 INTEGRATING LCA AND LCC

When considering LCA, decision-makers must also take the economic consequences of alternative products or designs into account. However, Norris (2001(b)) mentions that *“neither the internal nor external economic aspects of the decisions are within the scope of developed LCA methodology”*. Besides, by looking at the LCA methodology standards (ISO 14040 series), it can be stated that these standards do not address the integration of economic analysis combined with LCA.

The challenge of integrating LCA within LCC or vice versa lies within the significant methodological differences (Santos, Costa, Silvestre, & Pyl, 2019). The base of the gaps is because LCA and LCC are each intended to give answers to different questions, see paragraph 2.1. For example, the *life-cycle* of both methods is being addressed differently. According to Norris (2001(b)), some key factors are missing for the integration of LCC within LCA, which are:

- Cashflows that are related to investments in product or design changes;
- Costs and revenue streams which are not corresponding to the physical flows which are modelled within LCAs;
- The timing of cash flows considering the costs and benefits of the design and its present valuation;
- The risks of costs and their modification as an element of alternatives.

Altogether, integrating economic analysis within LCA requires not only the treatment of economic cost as another flow, but it requires the addition of a time dimension to the modelling dimension, the ability to utilise variables that have no causal dependence upon inventory flows, and the ability to deal with probabilistic situations (Norris, 2001(b)).

2.6.2 INTEGRATING LCA AND LCC WITHIN BIM

Several papers underline that integrating BIM with LCA and LCC can be considered as having excellent potential in providing timely information about a building's life-cycle environmental and economic performance. Still, the integration of BIM with LCA and LCC face several challenges that need to be considered with regards to the implementation within early design stages (Díaz & Antón, 2014; Santos, Costa, Silvestre, & Pyl, 2019). Although the integration of BIM-LCA is considered to establish a decrease of time and to implement environmental performance practicability of designs in early design stages, it must be stated that there are specific methodological difficulties of such an integration. There should be a general agreement for users of BIM, LCA and LCC frameworks to adapt this issue related to their requirements and preferences.

INTEROPERABILITY CHALLENGES

According to the literature, software integration can be seen as one of the most critical challenges, mainly due to the number of data requirements for BIM-LCA integration (Häkkinen & Kiviniemi, 2008; Díaz & Antón, 2014; Soust-Verdaguer, Llatas, & García-Martínez, Critical review of BIM-based LCA method to buildings, 2017). Soust-Verdaguer et al. (2017) identified that BIM-based LCA is constrained by time and the production and demolition of materials. A way to overcome this is by linking separate software via file exchange and using parametric formats (Häkkinen & Kiviniemi, 2008). As mentioned earlier, the LOD of an element represents the information content that can be used to perform environmental impact and cost calculations at a specific point in the design process.

Conversely, Soust-Verdaguer and García Martínez (2017) claim that due to the low LOD of elements modelled in early design stages, the LCA and LCC data input will be conditioned and even limited. Röck et al. (2019) concluded from different case studies that BIM-models varied significantly since most elements were modelled in LOD 100 to 200. In other words, building elements or sub-element levels were varying, such as wall elements that were modelled with separate structural, insulating and finishing layer. The application of LCA and LCC during early design stages should be established in a specific LOD that allows for quick modelling, and a user-friendly and reliable environmental and economic impact checking environment.

An overlapping challenge, mentioned by Rahmani Asl et al. (2015), is the interoperability work among the different software applications with regards to BIM and environmental performance. Although the use of an IFC-schema is efficient, various applicants of BIM within the AEC-industry use different software according to their work type and tasks in general. This difficulty results in data loss in the process of information exchange which causes unnecessary duplication of work for data restoration (Jalaei & Jrade, 2015). With regards to a BIM-based LCA and LCC integration, a drawback is that the current approaches are usually based upon that the BIM model remains linked to a specific software application (Santos, Costa, Silvestre, & Pyl, 2019). However, the design process requires input from various disciplines that make use of different software applications, presuming that they meet their specific needs. To be more specific, in the early design stages, a conceptual design is created in *SketchUp*, while the preliminary design is developed in *Revit*, for example (van Gemert, 2019). Although such a solution, in which a single BIM environment would inhibit an ongoing assessment process of the same design by different software applications, can be considered as ideal, the comprehension of such a process requires at least two software applications that can exchange and use data interoperable. Interoperability enables software applications to exchange information and practitioners to collaborate.

On the other hand, there is a challenge with regards to the open exchange standards since there still is a need for linking the BIM-model with external databases (Antón & Díaz, 2014; Bueno, Pereira, & Fabricio, 2018; van Gemert, 2019). Current approaches are mainly focussed on integrating LCA within the parameters of a model within the BIM-software packages. Besides, Nwodo et al. (2017) mention that one of the significant limitations of the current practice of LCA and LCC has involved *“the interoperability (data loss) issues, as well as the complexity arising from employing many tools especially at the early building design stage”*. Besides, considering the quantities that are generated from a BIM-model by IFC-schemas are usually not directly attached to objects. Instead, the quantities are written as property sets and therefore subsequently related to an object or object type. Depending on the software application, the quantities can be linked to a specific property set.

As previously discussed, there are three approaches to integrate BIM with LCA and LCC. These three approaches can be distinguished in two main streams (Díaz & Antón, 2014). One stream incorporates the assessment procedure within a single BIM environment, also known as *closed BIM*. The main drawback of these approaches that the BIM model remains linked to a specific software application. By looking at the different design stages, it can be stated that the BIM-model requires input from various disciplines which utilises (different) software applications.

Considering the early design stages, different software applications can be used in order to make a design. Yet, the lack of information within BIM-models not only lead to ineffective decision-making, but it also prevents automatic simulations of the project in different domains (Iacovidou, Purnell, & Lim, 2018; Santos, Costa, Silvestre, & Pyl, 2019). It is necessary to consider a solution in which integration is established within a single BIM-environment that provides the possibility to have a continuous assessment process of the same design by other software applications.

The other stream utilises open exchange standards to allow for environmental impact and cost assessment using various BIM software applications, also known as *open BIM*. The realisation of such a process requires at least two software applications to be interoperable with each other (Díaz & Antón, 2014; Santos, Costa, Silvestre, & Pyl, 2019). By using multiple applications through open BIM standards, the users are enabled to collaborate with other project participants (Soust-Verdaguer, Llatas, & García-Martínez, 2017; Bueno, Pereira, & Fabricio, 2018).

PROVIDING FEEDBACK

A common problem when applying LCA and LCC in a design process, with regards to providing feedback, is that in early design phases, the options for selecting various solutions are numerous. Furthermore, the available information on the products and materials within a design are scarce (CIRCE, 2013). LCA studies need detailed information, which results that these studies are usually performed during the completion of the building since at that stage that all the information is clear. This high standard limits the LCA-application in being only descriptive rather than providing feedback to improve the design (Jalaei & Jrade, 2015; Soust-Verdaguer, Llatas, & García-Martínez, 2017). An LCC-study requires easily accessible data in all stages of a building's lifecycle, where key numbers, statistic information, and experience data is valuable data. Increased use of LCC at various stages in the decision process also requires increased data exchange and storage possibilities (Krigsvoll, 2007).

On the other hand, during the different stages of a project, more decisions are taken, which enables the increased performance of LCA and LCC studies. However, the results of these studies with regards to alternative designs are then restricted. As previously mentioned, current BIM-LCA and LCC applications lack feedback, exclusively with regards to the *correct response feedback* approach. This way of working is considered inefficient, since modifying the design to achieve a high standard of environmental and economic performance criteria often results in backtracking (Schlueter & Thesseling, 2009; Santos, Costa, Silvestre, & Pyl, 2019). Although most applications provide feedback in terms of correctness of the results, these applications do not provide any guidance to practitioners in how to resolve issues (Nwodo, Anumba, & Asadi, 2017). Comprising approaches to get better information about alternative options early in the design process and to get faster predictions and calculations of rough results are required. More extensive use of BIM models enables the possibility of generating an automatic data flow between different databases, tools and BIM models. This integration will support the simplification and implementation of LCA and LCC in the AEC-industry (Krigsvoll, 2007).

Although some literature proclaims that valuable data has to be included within the BIM software model (Antón & Díaz, 2014; Santos, Costa, Silvestre, & Pyl, 2019), it must be stated

that in order to provide feedback with the use external databases is necessary. An approach that provides insight on how to combine environmental impacts to costs and how to optimise the decision-making process is linear programming (LP). While this method has not been researched to its full potential in the AEC-industry with regards to the integration of LCA with LCC, it can aid decision-makers to choose between different products or materials of design. Research in other fields of expertise has acknowledged this potential, in which the system or design can be optimised on different environment as well as economic objective functions (Azapagic, 1996; Azapagic & Clift, 1999; You, Tao, Graziano, & Snyder, 2011). Concerning the optimisation of the objective functions, it supports the identification of the best compromise solution for improving the performance of the system or design. Moreover, according to Azapagic (1996), optimising design through LP allows for complex interactions between different parts of a system or product. Within the methodology, this approach will be explained in more detail with regards to its potential relationship with BIM, LCA and LCC.

3 A BIM-BASED LIFE-CYCLE ENVIRONMENTAL AND ECONOMIC IMPACT ASSESSMENT TOOL

The previous chapter examined the importance of utilising BIM for environmental and economic assessment from a life-cycle perspective. Besides, it became clear that there is a critical need for integrating environmental and economic implications into assessment approaches in order to get a more holistic evaluation of environmental and economic impacts of a design. The availability of both environmental as well as economic performance can be affected if the trade-offs between the different aspects are used in a decision-making process. This availability may result in the selection of more sustainable product- or material choices, especially during early design stages. However, as demonstrated in the literature review, several challenges were identified that hold back the integration of environmental and economic Life-Cycle Assessment. A couple of these challenges that influence the decision-making process were the lack of integration methods, the high amount of data requirements and the lack of integrated LCA and LCC applications for the early design stages. This chapter seeks to find a solution to the combination of the techniques for integrating the environmental and economic performance of buildings in order to make a comprehensive impact assessment. The proposed solutions will form the base for the development of the application. This application will obtain the BoQ of all the building elements present in a BIM-model and link the data to an external database consisting of LCA and LCC data in order to calculate the environmental and economic impact of the proposed design by means of the proposed solutions. The results of this application will be presented in an user-interface, in which the different utilised building materials of the BIM-model are compared to other materials in order to optimise the design.

This chapter starts to show the requirements for integrating LCA and LCC through BIM. Once these requirements are presented, the base of the application is set up, consisting of the functional requirements as well as the technical content of the application. Afterwards, the system's architecture of the application will be further explained, starting by the elaboration of the content of the different python modules used to develop the application. In addition, the components of the system's architecture that form the core of the application are further discussed based on the IPO-method. Lastly, a list of improvements of the application is provided based on the feedback of experts from the AEC-industry.

3.1 REQUIREMENTS FOR INTEGRATING LIFE-CYCLE ENVIRONMENTAL AND COST ASSESSMENT WITH BIM

Based on the literature research, this section will combine the different requirements needed to perform an LCA and LCC analysis through the use of BIM.

3.1.1 THE STUDY APPROACH

As mentioned in the literature research, LCA and LCC studies can be subdivided into four studies: screening, simplified, streamlined, and complete. Both the screening and the simplified studies can be used for an initial assessment of the environmental impacts of buildings or products, based on its project stage. However, designs change a lot during the early design stages, which results in that specific data (e.g. building systems, material choices) are not established yet during these stages. During the performance of a screening and simplified study, this data needs to be available. Otherwise, these studies cannot be executed. A streamlined study is, therefore, most suitable for the early design stages since it functions as an ad-hoc assessment approach in which a quick and comprehensive assessment can be made based on generic data. With regards to a complete LCA and LCC-study, it would be necessary to insert specific information of the project manually. Required project details vary from the building's service life, transportation aspects, and used utilities and waste which is generated during the construction and operational phase.

By looking at the AEC-industry, the foundation of quality BIM-models is laid within the early design stages. Considering the many design changes in the early design stages, it is not always possible to perform any of the data-required studies, which are the screening, simplified and complete studies. Therefore, a streamlined study is an ideal approach to perform an LCA and LCC assessment in early design stages because this study is performed using generic LCA and LCC data. The generic LCA data consist of different indicators mentioned in Table 1. With regards to the generic LCC data, as mentioned in section 2.2.2, only the acquisition costs will be used for the assessment of LCC.

3.1.2 THE INTEGRATION APPROACH

Considering the integration of LCA and LCC through BIM, three integration approaches can be utilised (section 2.4). Santos et al. (2019) proclaim that having a BIM-model would enable automatic streamlined LCA and LCC analysis of a project since different generic parameters of a BIM-model can easily be linked to LCA and LCC data. Considering the three integration approaches, this means that only integration approach two and three can be used since they directly use BIM-data to enable LCA and LCC calculations. To follow the third integration approach, the BIM-software packages should have libraries that consist of matching BIM-data containing LCA and LCC data. The data should be interpreted by other BIM-software packages. Otherwise, discrepancies and incomplete calculations occur, which result in deficient LCA and LCC analysis. Santos et al. (2019) researched the third integration approach, which initiates the inclusion of LCA and LCC within a BIM-based environment. Including LCA and LCC data within a BIM-model enables the user to automatically calculate the results of LCA and LCC analysis within a single BIM-software package. Although a streamlined study could be done automatically, Santos et al. (2019) concluded that the third integration approach was considered to not yet be optimal. The researchers mentioned that the information in building elements still lacked comprehensiveness due to the fact that an IFC schema does not contain suitable properties to store the type of information required by each environmental impact category.

By looking at the entire design process, in general, it can be stated that project participants use different BIM-software packages to make their models. Following the second integration approach, a BoQ needs to be extracted from a BIM-model and can be utilised to make calculations based on the available data in an LCA and LCC database. In other words, the input from the used BIM-model comprises the BoQ of the physical model, which is linked to an external LCA and LCC database. By having one overall LCA and LCC database that can be linked with BIM-data, multiple stakeholders with various BIM-software packages can collaborate together and execute LCA and LCC analysis. By following the second integration approach, the libraries within the BIM-software packages only need to contain matching classification codes referring to the different building elements. The only requirement for project participants is that they need to utilise a BIM-software package that allows the user to specify the different building elements with classification codes, such as Revit, Sketchup or ArchiCad.

The most significant difference between the approaches is that the second approach uses an external database that functions as the input of material characteristics with regards to their environmental and economic nature, whereas a BIM-model in the third approach already contains LCA and LCC data. An external database allows multiple project participants to collaborate based on a mutually agreed classification system. Altogether, for the development of the application, the second integration approach is used for the calculation of the environmental and economic impact scores. This decision is made based on the conclusion of Santos et al. (2019), the different BIM-software users in a project, and the requirement of only having a similar classification system for classifying building elements in BIM-models.

3.1.3 THE DATA REQUIREMENTS

The minimal requirements for data input for the environmental and economic assessment of a design by using LCA and LCC and IFC are presented in Figure 14. These minimal requirements are based on the environmental-economic performance structure by Ahmed and Tsavdaridis (2018), the proposed framework of Santos et al. (2019), combined with the available environmental and economic as well as IFC-data mentioned in the literature research. It must be noted that a differentiating aspect between the requirements for environmental and economic impact assessment from Santos et al. (2019) and the proposed requirements is that the proposed list of environmental indicators is more extensive. This extensiveness originates from the Dutch environmental regulations, which implies to use a more extensive list of environmental indicators.

With regards to the matching classification, an essential requirement in the list of requirements is the recurring assembly code of a building element or material. This code enables the linkage between the IFC-data, LCA data and the LCC data. If this code is not present or incorrectly formulated in either the external LCA and LCC database or the IFC-model, the calculations with regards to the environmental and economic impacts cannot be made. Besides, the assembly codes need to be based on an identical classification scheme. Within the Dutch AEC-industry, the NL/Sfb classification system is used as standard. The coding system of the NL/Sfb divides a building into products that are subsequently divided into different categories. The main structure of the coding system contains three groups of two digits each, in which the first group of digits denotes the location of the building element, such as internal walls (22) and floors (23). The second group of digits indicates the function of the building element. The final group of digits refers to the materialisation of the building element.

Requirements for a streamlined LCA and LCC analysis		
Required Environmental information (1) Eleven indicators determining the environmental impact - Global warming potential - Stratospheric ozone depletion potential - Photochemical oxidation potential - Acidification potential - Eutrophication potential - Human toxicity potential - Terrestrial ecotoxicity potential - Freshwater aquatic ecotoxicity potential - Marine aquatic ecotoxicity potential - Abiotic depletion potential of composites and minerals - Abiotic depletion potential of fossil fuels (2) Assembly codes of building elements or materials Overall Environmental Calculation Based on data from the IFC-model, corresponding to matching assembly codes. $EI^{MC} = \sum_{a=1}^i (Q_a^M \times EI_a^M)$	Required IFC-data (BIM-model) (1) Type of building element - IfcWall, IfcSlab (2) Assembly codes of building elements or materials - IfcRelAssociationClassification or IfcPropertyset (3) Measurements of building elements or materials - Areas (IfcElementquantity or IfcPropertyset) - Volumes (IfcElementquantity or IfcPropertyset) - Lengths (IfcElementquantity or IfcPropertyset) - Heights (IfcElementquantity or IfcPropertyset) - Widths (IfcElementquantity or IfcPropertyset) (4) Gross floor area of building - IfcSpace or IfcElementquantity Optional IFC-data (BIM-model) (5) Reference for variations of matching assembly codes - Material-reference (IfcPropertyset)	Required Economic information (1) Acquisition costs - Cost of materials - Costs to build the building elements (2) Assembly codes of building elements or materials Overall Economic Calculation Based on data from the IFC-model, corresponding to matching assembly codes. $C^{MC} = \sum_{a=1}^i (Q_a^M \times AC_a^M) + \sum_{b=1}^j (Q_b^M \times C_b^{C2B})$

Figure 14 - List of requirements for calculating LCA and LCC based on IFC data

3.2 THE BASE OF THE APPLICATION

The requirements for integrating LCA and LCC through BIM has been explained in the previous section. These requirements form the base of the application, meaning that the functionality of the application depends on the input of the LCA, LCC as well as the BIM data. Elaborating on these requirements, the base of the application is discussed in this section.

3.2.1 FUNCTIONAL REQUIREMENTS OF THE APPLICATION

With regards to the literature research, key focusses from LCA-LCC integration with and without BIM have been identified. By executing unstructured interviews with different experts in the AEC-industry, the different functional requirements identified in the literature research are validated and extended. These meetings have been performed in an informal setting in order to stimulate out-of-the-box thinking and logical reasoning from the perspective of the experts' field of expertise, which means that no formal notes of the meetings were made. A list of the different interviewees is presented in Appendix II.

All requirements of the applications can be considered as significant. However, they are ranked in order of priority in order to deliver the most considerable benefits (Kossiakoff, Sweet, Seymour, & Biemer, 2011). A systematic approach that structures the prioritisation process is the *MoSCoW method*. The term *MoSCoW* itself is based upon the four prioritisation categories; (i) **Must**-haves; (ii) **Should**-haves; (iii) **Could**-haves; and (iv) **Would**-haves. This method is a principle that is widely used in management, business analysis, and software development in order to prioritise the requirements of new developments. Initially, all the requirements should be met. However, when the delivery timescale of the application is threatened, the first to be removed requirements are within the **could** and **would**-have requirements.

Must-haves include the core functionality and features of the product. These requirements form the components for the *minimum viable product* (MVP) (Parker, 2019). With regards to the application, the application **must** contain the following functional requirements:

- Process data from the BoQ of an IFC-model which is stored in the IFC-schema (IFC 2x3);
- Process data extracted from an external database, containing LCA and LCC data;
- Link data from the IFC-model to the external database;
- Generate environmental and economic impact score of a streamlined LCA/ LCC-study;
- Generate feedback that aids users to make well-considered environmental and economical design decisions in the early design stages.

Should-haves are essential for the development of the product but are not considered to be a critical component for the base functionality of the product (Parker, 2019). In other words, the product would still be functional if these requirements are not implemented. For the development of the application, the following functional requirements **should** be present:

- Process data from an IFC-model which is stored in the IFC-schema (IFC 2x3);
- Accept IFC-files from different BIM-software as input for the application;
- An easy-to-use graphical user interface.

Could-haves are desirable requirements, which could improve the user experience, for example. They should only be implemented if time and resources permit the development of these requirements (Parker, 2019). The following functional requirements **could** be:

- Generate environmental and economic impact scores for individual building materials;
- Provides the option to include the personal preferences of the users.

Would-haves are requirements that will not be implemented in the current version of the product, but would not harm this product in any way (Parker, 2019). The following functional requirements **would** be implemented in future versions of the application:

- IFC versions beyond IFC 2x3 or other extensions of the IFC-file format;
- Generate an environmental and economic impact score of a complete LCA/ LCC-study;
- Generate feedback that aids users to make well-considered environmental and economical design decisions in all design stages;
- Develop a web-based application to support accessibility by users at any place.

3.2.2 THE TECHNICAL CONTENT BEHIND THE FUNCTIONALITY OF THE APPLICATION

There are many ways to develop an application. Based on the different unstructured interviews (Appendix II), the development of the application should be presented in an easy-to-use environment which enables any user to make an LCA and LCC analysis. Possible interfaces of the application could be a simple spreadsheet, a web-based application or a stand-alone application. The proposed interface of the application was based on the programming skill set of the author at the start of the research. It was found not feasible to create a web-based application. As such, it was decided to develop a stand-alone application.

The core of the application development is a Python script. The version of the programming language used to develop the application was Python 3.5. This programming language is selected as the preferred language for this development since its syntax is easy to comprehend and implement. Furthermore, it offers flexibility in extracting and creating data using numerous third-party packages in order to extend the functionality of the programming environment. Table 5 provides an overview of used Python packages and its explanation.

Table 5 - Python packages that have been used for the development of the application

Package	Version	Type	Function
PyQt4	4.13.1	GUI Toolkit	Set of Python bindings of cross-platform GUI toolkit Qt that provides a standard graphical user interface.
Pandas	0.25	Software library	Provides robust data structures for analysis.
NumPy	1.17.3	Software library	Adds support for large, multidimensional matrices, along with an extensive collection of high-level mathematical functions to operate on the arrays.
Matplotlib	1.5.3	Plotting library	that offers a wide variety of plots. Generally, it is used as an extension of NumPy.
IfcOpenShell	3.5	Software library	Aids users to work with the IFC-file format.
.xlsxwriter	1.2.6	Package	Writes text, numbers and formulas to multiple worksheets in .xlsx-format.
Importlib	1.3	Package	Functions as an import mechanism for (re)loading codes in packages and modules.
PythonOCC	0.18.1	Software library	Provides features such as geometrical operations, advanced topological, and data exchange.

3.3 THE SYSTEM'S DESIGN

Combining the requirements for data-input with the content and functional requirements of the application, a system's architecture of the application was set up. This system's architecture can be described as an information flow that proceeds from the input from a BIM model and LCA and LCC database to the potential output that users need in order to make well-considered decisions for the design. As can be seen in Figure 15, the system's architecture follows a development process of the application that matches the IPO-method. The internal processes within the application have a direct influence on the process, whereas the external processes are executed aside from the application. To be more specific, the application provides insight and feedback on the results of the environmental and economic impacts based on the BoQ (*internal process*), and it supports the user to make environmentally and economically considered decisions which can be used to optimise the design (*external process*).

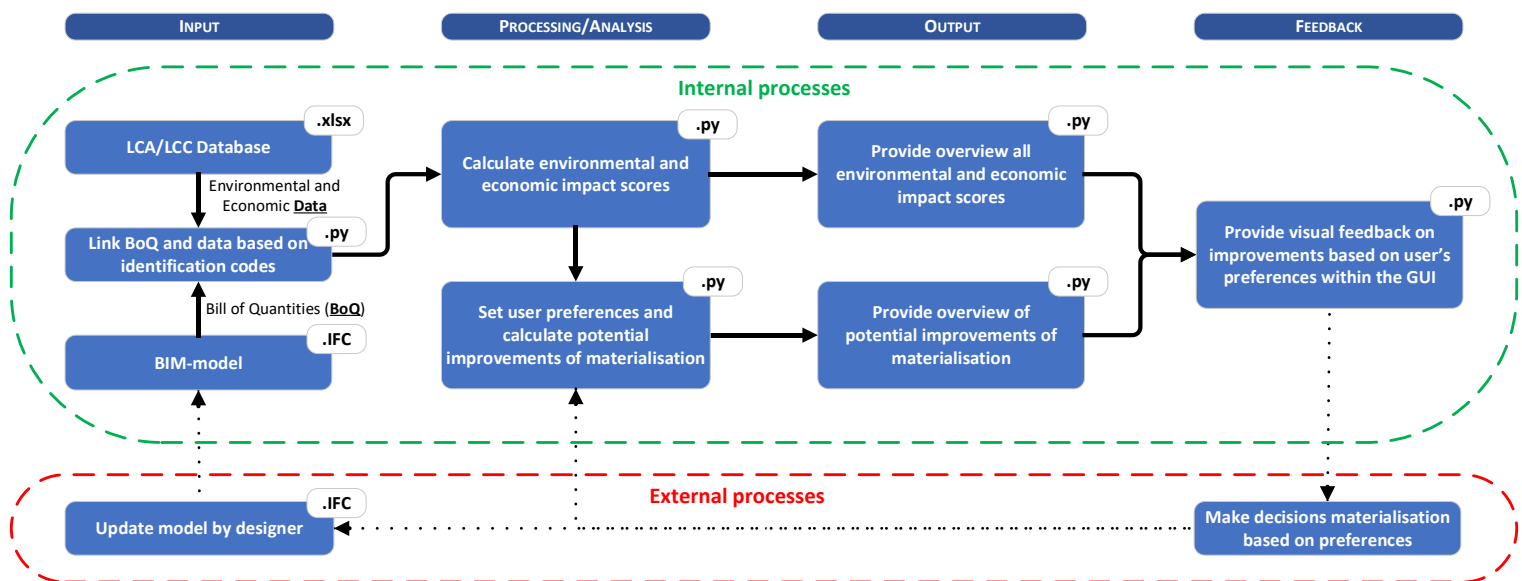


Figure 15 - System's architecture of the application

For the development of the Life-cycle Environmental Impact and Cost Assessment application, from this point on called the LEICAS-application, a variety of modules are linked to each other. Based on the system's architecture, different modules are developed, which together form the core of the system. The reason why there are multiple modules are not implemented into one main module, is that the total script would be extensive in terms of length. In order to keep the modules run smooth, it is recommended to keep the Python code small and understandable (Álvarez Cid-Fuentes, et al., 2019; Sagar, 2019). Therefore, the decision was made to make multiple modules, all with different functionalities. The workflow of how the different modules are interlinked with each other is presented in Figure 16.

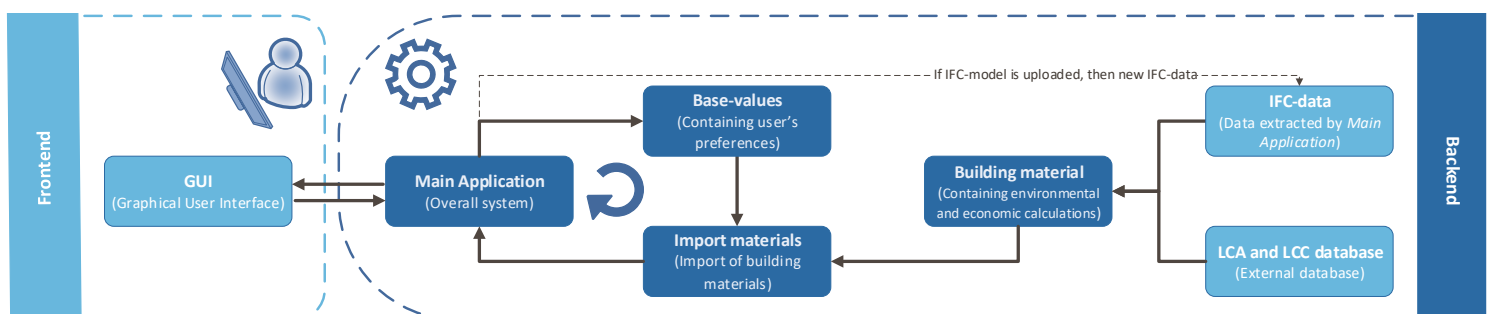


Figure 16 - Workflow of the different modules of the application

Overall, it can be stated that there is a distinction in visibility towards the user. The visible part for the user of the application, the so-called frontend, is formed by a graphical user interface (GUI). The GUI enables the user of the application to comprehend the results of the calculations. The non-visible part of the application, the backend, is developed to make calculations based on the various data inputs. In order to have the frontend presenting the results, a linkage is made with the backend.

Within the backend of the LEICAS-application, four modules have been developed: (1) the *main application* module; (2) the *base-values* module; (3) the *import materials* module; and (4) the *building material* module(s). With regards to the *building material* module, for each material within the LCA and LCC database, there is a separate module.

Furthermore, the backend of the application also contains the two data sources, the external database and the IFC-data, which are key for the environmental and economic calculations. Although they consist of valuable data, the main core of the application lies within the four mentioned backend modules. In order to comprehend the content of these different backend-modules, these modules will be further discussed. Since the full scripts are extensive, the scripts can be found in Appendix III.

3.3.1 THE MAIN MODULE

The *main* module contains the functionality of the different preferences of the user and links the *GUI* with the scripts that make the calculations. The *main* module forms a bridge between the different environmental and economic calculations and the presentation of these results to the user within the *GUI*.

In addition, the *main* module enables the import of an IFC-model. Since the different underlying scripts are based on the data from the IFC-model, the script of the *main* module provides an extraction of the data of the IFC-model and exports it into a .xlsx-format. With regards to the data from the IFC-model, the data will be extracted from the IFC-model and will be translated into a .xlsx-file format. A couple of researches state that transparency in data usage and calculations is highly recommended (Guinée, 2016; Iacovidou, Purnell, & Lim, 2018; Schmidt, 2018; Santos, Costa, Silvestre, & Pyl, 2019). Consequently, if an application only extracts the data and uses the data only for calculating, the user cannot check if the IFC-content is correct and, in general, does not know what data is used for the calculations. Therefore, the chosen form of extracting and processing the IFC-data enables the user to check the content of the IFC-model within the .xlsx-file.

Since the application must process a lot of data to provide environmental and economic calculations, the links between the different modules need to be updated. These updates are needed because the information used to make the calculations must be similar within the different modules. As can be seen in Figure 16, the different modules within the application require information from each other. Through the use of the *importlib* package, the different modules are updated while working in the application. In other words, if one aspect within the application changes, every other module is updated. In order to get a list of specifications related to the required project-specific information and the user's preferences, a part of the *main module* script provides another *module* script that summarises this list of specifications. This other *module* script is the *base-values* script.

3.3.2 THE BASE-VALUES MODULE

By running the *main module* script, the *base-values* module will automatically be exported as Python-script and will function as an essential base for the other scripts. The *base-values* module provides a summation of all the preferences the user has as input in the application. When the *base-values* module is updated, all the other scripts provide results that are presented in the application. As can be seen in Figure 17, the base-values script is subdivided into two parts. The first part of the module provides information about the project-specific information, which is based on the building type of the design, and the lifespan of the building.

```
### Base-values ###  
building_type = "Residential"  
LDg = 75  
  
calculation_choice = "Generic"  
building_system = "All"  
factorenvimpact = 0.5  
factorecoimpact = 0.5
```

Figure 17 - Snippet of *basevalues.py* script based on a selection of decision criteria

The second part of the module is based on the user's preferences. Starting with the calculation choice and the building system, it can be stated that these factors function as a key for the input of the different materials within the *import materials* module. For example, these factors enable the application to calculate the environmental and economic impact based on the type of building system the user wants to have. By having a selected type of building system, the application filters the requested materials related to the building system and provides calculations based on these requested materials.

The other two user preferences are related to the importance of environmental and economic impact. These two factors are based on the preference of whether the user wants an environmentally considered advice or a more economic considered design and form the base of the feedback provided by the application. Together, these two factors have a score of 1. In the case the user prefers to have balanced feedback, these factors will be both 0.5. Based on the user's preference, the value of the factors can shift to the required importance of impact.

Further clarification of the content of the *base-values* module is given in section 3.5.

3.3.3 IMPORT OF BUILDING MATERIALS MODULE

Once the different preferences of the user are set, the application makes an import of the different building materials based on the preferences mentioned in the *base-values* module. As have been mentioned before, this module filters the different materials corresponding to the different preferences. By selecting a certain building system or a calculation choice, the user provides the system information on what level of detail the calculations should be made. These calculations are related to the various *building material* modules and to which materials should be imported into the application. The function of the *import materials* module is that the module gathers the environmental and economic impact of the different building materials based on the user's preferences and provides a link between the *main application* module and the different *building material* modules.

3.3.4 VARIOUS BUILDING MATERIAL MODULES

The various building materials form the keystones of the application. Within the module of the *building material*, all the calculations with regards to the environmental and economic impact are made. As can be seen in Figure 16, the *main application* script extracts the data from the IFC-file and translates the IFC-data into a .xlsx file format. The extracted IFC-data, combined with the external LCA and LCC database, is then used as a base for the calculations of the environmental and economic impact of a certain building material. The extensiveness of the number of *building material* modules depends on the number of building materials provided by the LCA and LCC database.

3.4 THE INPUT

At the start of the calculation of environmental and economic impact assessment, the application requires two data sources as input. First of all, there needs to be an input from an external database with LCA and LCC data. The Dutch NMD is used as input for the required LCA data within the database. This NMD-database is required to make calculations for the building, following the Dutch legislation. As mentioned earlier, MPG 3.0 was launched in July 2019. However, at the moment of writing this thesis, the NMD-database has not yet been updated. The developers of the NMD proclaim that the current version of the NMD is comparable to the new version (Milieudatabase, 2019), and will be made available early 2020. Therefore, the calculations used by the NMD 2.3 version are used for the calculations of the environmental impact scores.

The environmental characteristics of building products are documented in so-called product cards within the NMD database,. These different product cards have been set up with regards to the *Environmental Product Declarations (EPDs)* – see Appendix IV. The categorisations of the NMD can be subdivided into three categories: (1) *Category 1* – proprietary data from manufacturers that are tested, (2) *Category 2* – non-proprietary data that are tested, and (3) *Category 3* – non-proprietary data that are not tested. The second and third category can also be characterised as being general LCA data, since these have not been linked to specific products of manufacturers. Since the first and second category of data are not freely accessible, *Category 3* will be used for the development of the application. The LCC-data in the database will be used in order to determine the acquisition costs of the building that are needed to perform a streamlined LCC analysis (Santos, Costa, Silvestre, & Pyl, 2019). Since the focus of this research lies within the early design stages and considering the streamlined LCC-study, the only requirement for the LCC database, is that it entails the initial material costs and the construction costs of the materials. Therefore, the LCC database will be *ArchiCalc*. This database is comprehensive, and consists of multiple variations of the same materials, i.e. widths or colours. The application must retrieve the different data in order to link the different components; LCA, LCC and BIM. In the next sections, the data retrieval of the three main components and the link between the LCA and LCC data within a database will be discussed.

The second data source that is needed as input, is the input of the IFC-model. As already mentioned in Figure 14, in order to make LCA and LCC calculations, the minimum content of this IFC-model is based on the minimal requirements for calculating the environmental and economic impact of a design. The minimum content of an IFC-model must consist of at least the assembly codes, areas and widths of building elements. In the case that the IFC-model contains materials with the same assembly codes but with different widths, the material references are required as well. If there are no building elements that have materials with

material preferences present, the categorisation of these elements will be based on the assembly codes. This section will elaborate further on the retrieval, the processing and the linkage of both data sources.

3.4.1 DATA RETRIEVAL FOR LCA AND LCC

Multiple alternatives can be used in order to retrieve and map the data from the different databases that were used for the development of the application. The most reliable option, in which data loss is minimised, is linking the databases to a so-called Application Programming Interface (API). An API is a set of definitions in the form of an interface through which a computer program can communicate with another program or component, usually in the form of one or more libraries. In other words, the data owner of the database(s) must provide access to their API in order to get full access to the data. However, by looking at the databases chosen for the development, it can be stated that it is highly unlikely that the data providers would grant full access to their databases.

A way to overcome this is by directly collecting the data from the source itself. For the data collection of this research, this means that the data from the databases' websites will be collected automatically. This method, also known as data crawling or web scraping, uses a code which scours the source in order to find different data that can be used for other purposes, such as calculations and analysis. A turning point on data crawling is that it is tough to determine the legality since data crawling, especially web crawling, can be used for malicious purposes. Moreover, by the abusive behaviour of requesting large amounts of data, data crawling can lead to server crashes. Therefore, data crawling is considered to be a questionable method. The easiest way of retrieving data from the different databases is by manually collecting the data from these databases and storing it in a machine-readable file format. However, the chance of data loss is higher by manually collecting the data due to human failure; the combination of multiple databases into a single database is required for the generation of feedback. The proposition is to combine the LCA data and the LCC data into one single. This integration enables the increasement of accurate comparisons that results in the optimisation of design through its lifecycle. In order to guide this combination, it is necessary to have a standard classification system.

3.4.2 DATABASE SETUP

As mentioned earlier, a standard classification system is necessary to be introduced to identify the different products in the product database. The convenience of the utilised LCA and LCC data sources is that they already use the NL/Sfb classification system. Since both LCA and LCC data sources already use the NL/Sfb classification system, this system will be used to classify materials, such as wood and concrete, related to their building elements, such as walls and floors. However, a notion must be made on the content of the sources for the LCA and LCC data with regards to the NL/Sfb classification system. Although both data sources make use of the NL/Sfb classification, variations in coding occur. For example, within the NMD-database, the building element *load-bearing walls* is classified with the generic building assembly code "28.01", whereas within the *Bouwkosten-online* database, these same wall types are classified by generic building assembly code "21.21". Since these discrepancies exist, it is necessary to manually select the different building elements and their related materials within the data sources with the same description in order to enable well-grounded feedback. Keeping in mind that both data sources have discrepancies, the proposed database was set up manually using

Microsoft Excel as a working environment. Each building element within the database was given a unique identifier so that the Python script able to retrieve data of various materials within building elements. The data entries were checked thoroughly in order to reduce the possibility of irregularities among the LCA and LCC data.

The database consists of seven tabs. The *product* tab, contains the overall LCA-content of the database. In other words, this tab refers to the required LCA-data of materials based on the assembly codes and contains generic data of individual products related to the LCA-data. The *components* tab, contains the generic data of the different components of materials containing the LCA-data. This means that the different materials consist of one or more components. This tab organises the different components related to these materials. The last layers of the LCA-data are provided in tabs three to five. The information in these tabs contain the LCA-data related to the different components of the materials. The so-called “*tc*”, “*pc*” and “*ac*” tabs contain environmental data on the life-cycle impact of transportation (*tc*), production (*pc*) and waste processing (*ac*) and are related to the eleven environmental impact categories mentioned in Table 1. These impacts are classified by different codes that are related to the different LCA-data. The content of the LCA-data within the database is presented in Figure 18. For example, the linkage of the LCA-data of the material consisting of assembly code “22.03.001” is indicated within the highlighted areas.

Figure 18 - Tabs in the database consisting of LCA data

The sixth tab contains the generic LCC-content of the database and links the different LCA assembly codes to the LCC assembly codes. In this tab, the cost impact will be based on the

mean of the different building components. The reason for calculating the mean of this component is because, within the utilised costing database, the variety of individual elements make it difficult to estimate the actual value of the final cost. Taking the brick wall category as an example; this category contains ten different brick wall types. The prices of these bricks vary in a particular range. Therefore, in order to make a specific estimation of the costs involved in the early design stages, the mean cost is used. These price variations per building material are mentioned in the last tab, the *costcomponents* tab.

	A	B	C	D	E	F	G	H	
1	building	product	explanation	bouwkostenonline code	unit	mcm	min mcm	mean mcm	max mcm
2	22.03.001	Gips met vlaskern	22.13.11	m2	44.34	44.34	44.34	44.34	
3	22.03.002	Poreuze baksteen	22.21.12	m2	84.51	92.96	100.04		
4	22.03.003	Lichtgewicht keramisch	22.21.15	m2	68.9	74.79	83.34		
5	22.03.005	Keramische binnenmuur	22.21.16	m2	48.53	64.14	86.5		
6	22.03.016	Licht Leemsteenmetseelwerk	E_22.03.016	m2	120.28	166.82	221.78		
7	22.03.024	Baksteenmetseelwerk	22.21.12	m2	84.51	92.96	100.04		
8	23.01.003	Balk en broodjes; pref	23.21.51	m2	43.18	43.76	44.34		
9	23.01.004	Europees naaldhouten	23.21.02	m2	52.65	55.72	58		
10	23.01.012	Staalframe element	23.21.92	m2	72.43	79.47	84.72		
11	23.01.013	Staalplaatbetonvloer	23.21.92	m2	72.43	79.47	84.72		
12	23.01.015	Massief houtenvloer	23.21.01	m2	181.96	181.96	181.96		
13	23.01.054	Appartementenvloer	23.21.22	m2	71.19	79.73	87.76		
14	23.01.072	Beton, in het werk ges	28.21.22	m2	121	121	121		
15	28.01.003	Baksteen geperforeerd	22.21.16	m2	48.53	64.14	86.5		
16	28.01.028	Stampleem	E_28.01.028	m2	113.9	227.87	341.8		
17	28.01.029	Baksteenmetseelwerk	21.12.15	m2	90.13	98.58	105.66		
18	28.01.034	Beton, in het werk ges	28.21.12	m2	180.2	180.2	180.2		

	A	B	C	D	E	F	G	H	
1	bouwkostenonline	Explanation	hvh	load-bearing?	unit	mcm	min mcm	mean mcm	max mcm
36	22.21.12.807	Rood gelijnd	1	X	m2	€	-	€ 86.23	€ -
37	22.21.12.811	Wit gelijnd	1	X	m2	€	-	€ 97.79	€ -
38	22.21.12.812	Witroze gelijnd	1	X	m2	€	-	€ 85.84	€ -
39	22.21.12.817	Bruinzwart gelijnd	1	X	m2	€	-	€ 92.01	€ -
40	22.21.12.818	Zwart bezand gelijnd	1	X	m2	€	-	€ 100.04	€ -
41	22.21.15					€	68.90	€ 74.79	€ 83.34
42	22.21.15.201	Lichtgewicht baksteen 210x70x100mm	1	X	m2	€	-	€ 68.90	€ -
43	22.21.15.202	Lichtgewicht baksteen 210x100x50mm	1	X	m2	€	-	€ 76.59	€ -
44	22.21.15.203	Lichtgewicht baksteen 210x100x70mm	1	X	m2	€	-	€ 73.42	€ -
45	22.21.15.204	Lichtgewicht baksteen 210x100x100mm	1	X	m2	€	-	€ 71.72	€ -
46	22.21.15.205	Lichtgewicht baksteen 210x140x100mm	1	X	m2	€	-	€ 83.34	€ -
47	22.21.16					€	48.53	€ 64.14	€ 86.50
48	22.21.16.201	Keramisch metselblok 33x100x155mm, geperforeerde binnenmuursteen 18n/	1	X	m2	€	-	€ 58.57	€ -
49	22.21.16.202	Keramisch metselblok 33x120x155mm, geperforeerde binnenmuursteen 18n/	1	X	m2	€	-	€ 61.25	€ -
50	22.21.16.203	Keramisch metselblok 33x140x155mm, geperforeerde binnenmuursteen 18n/	1	X	m2	€	-	€ 64.69	€ -
51	22.21.16.204	Keramisch metselblok 33x140x155mm (wooningscheidend), geperforeerde bini	1	X	m2	€	-	€ 72.90	€ -
52	22.21.16.205	Keramisch metselblok 300x200x155mm, geperforeerde binnenmuursteen 18n/	1	X	m2	€	-	€ 71.70	€ -
53	22.21.16.206	Keramisch metselblok 33x100x155mm, geperforeerde binnenmuursteen 21n/	1	X	m2	€	-	€ 59.32	€ -
54	22.21.16.207	Keramisch metselblok 33x120x155mm, geperforeerde binnenmuursteen 21n/	1	X	m2	€	-	€ 63.43	€ -
55	22.21.16.208	Keramisch metselblok 33x140x155mm, geperforeerde binnenmuursteen 21n/	1	X	m2	€	-	€ 66.22	€ -
56	22.21.16.209	Keramisch metselblok 33x140x155mm (wooningscheidend), geperforeerde bini	1	X	m2	€	-	€ 69.45	€ -
57	22.21.16.205	Keramisch metselblok 300x200x155mm, geperforeerde binnenmuursteen 21n/	1	X	m2	€	-	€ 86.50	€ -
58	22.21.16.206	Keramisch metselblok 33x100x253mm, geperforeerde binnenmuursteen 21n/	1	X	m2	€	-	€ 48.53	€ -
59	22.21.16.207	Keramisch metselblok 33x120x253mm, geperforeerde binnenmuursteen 21n/	1	X	m2	€	-	€ 51.30	€ -
60	22.21.16.208	Keramisch metselblok 33x140x253mm, geperforeerde binnenmuursteen 21n/	1	X	m2	€	-	€ 54.41	€ -
61	22.21.16.209	Keramisch metselblok 33x140x253mm, geperforeerde binnenmuursteen 21n/	1	X	m2	€	-	€ 69.67	€ -
62	22.21.01					€	181.96	€ 181.96	€ 181.96
63	22.21.01.507	Massief houten vloerelement [overspanning 6,10m], dikte 201mm, 7-lagen	1		m2	€	-	€ 181.96	€ -
64	22.21.02					€	52.65	€ 55.72	€ 58.00
65	22.21.02.101	Houten balklaag 75x150mm, hoh 600mm	1		m2	€	-	€ 55.71	€ -
66	22.21.02.102	Houten balklaag 75x175mm, hoh 600mm	1		m2	€	-	€ 56.43	€ -
67	22.21.02.103	Houten balklaag 75x200mm, hoh 600mm	1		m2	€	-	€ 58.00	€ -

Figure 19 - Tabs in the database consisting of LCC data

The content of the LCA-data in the database is represented in Figure 19. For example, the linkage of the LCC-data of the material consisting of assembly code “22.03.005” is indicated within the highlighted areas. In addition, the linkage of the LCC-data with the LCA-data is displayed in the first and third row of the left figure. The first row indicates the assembly codes related to the LCA-data, whereas the content of the third row indicates the assembly codes of the LCC-data.

3.4.3 BIM-INPUT

To elaborate on the content of the IFC-data: different property set types of the IFC-data can contain quantities, such as the *IfcElementQuantity* and *IfcPropertySet*. By extracting the quantities from the *IfcElementQuantity* property set the different object's physical characteristics. These characteristics can be linked to an object through the *IfcRelDefinesByProperties* relationship. This linkage has as the main benefit that it enables the possibility to calculate an object's surface area or volume. The content of this *IfcElementQuantity* can vary from just the surface area to attributes such as the length and width of the attribute. Entities within the *IfcElementQuantity* such as lengths and widths are stored within the *IfcQuantityArea*.

However, a notion must be made, since not every BIM software application allows for the *IfcElementQuantity* entity to be exported (van Gemert, 2019). An example is the use of Sketchup, in which the properties of attributes are stored within the *IfcPropertySet*. This property set can only be accessed through *IfcRelDefinesByProperties*. On the other hand, the *IfcPropertySet* can hold up various types of properties, whereas the one type of property that stores properties is the *IfcPropertySingleValue*. This type of property assigns a certain value of a building element, such as a measurement or an assembly code of a certain building element. By using the *IfcPropertySet*, the building elements within an IFC-model can be categorised by their assembly codes.

3.4.4 LINKING DATA SOURCES FOR CALCULATING THE ENVIRONMENTAL AND ECONOMIC IMPACT SCORE

For the calculation of the environmental and economic impact score, a link between the database with the BoQ of the IFC-model must be made. Figure 20 shows a part of the system's architecture, which explains the integration of the BoQ with the LCA and LCC data. Based on the identification code, following the NL/Sfb, and the calculation rules for both the LCA and LCC, the environmental and economic impact of a design can be calculated.

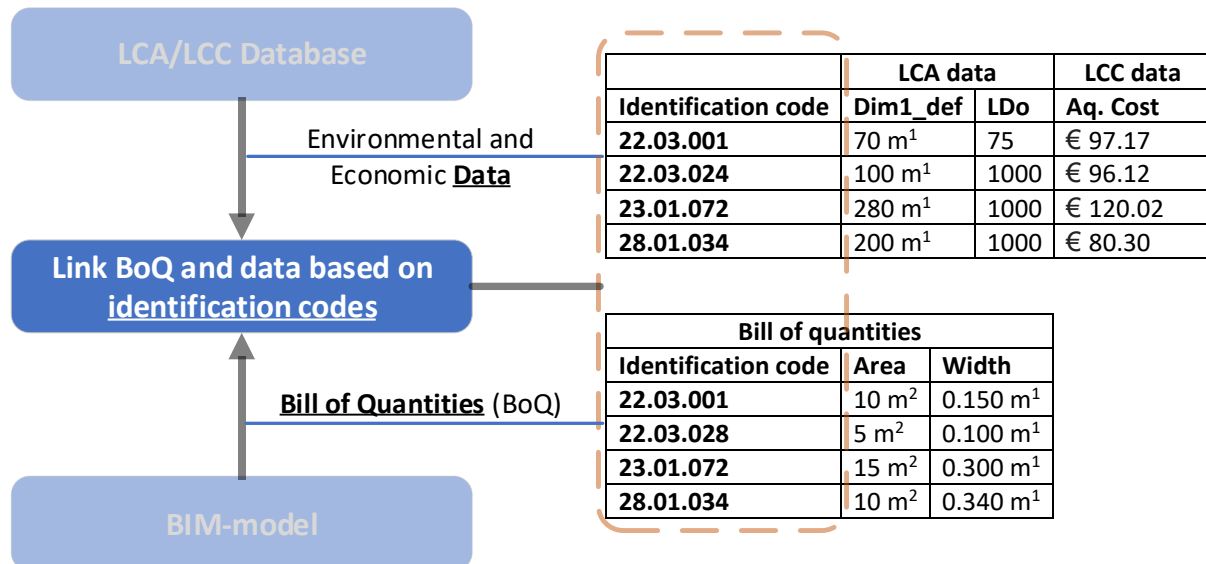


Figure 20 - Zoomed view of the system's architecture: combining BoQ-content with LCA and LCC data

3.5 THE PROCESS

During the process stage, an analysis will be made based on the data from the LCA and LCC database and the data from the IFC-model. For the calculation of the environmental impact, the LEICAS-application integrates the calculation rules considering the MPG-methodology in order to calculate the environmental performance of building products over their entire life-cycle. The LEICAS-application uses the initial costs of the building materials and construction costs of these materials in order to calculate the economic impact.

Furthermore, the analysis is based on project-specific information and different user preferences. With regards to the input of project-specific information, the user has to provide the lifespan of the building. Considering the user preferences, the user has to provide information based on the type of calculation, the type of building system, as well as the preference of impact importance of the design. These four aspects of the application will be discussed.

3.5.1 TYPE OF BUILDING

First, the user decides which building type the design is, since the lifespan of the different building types is required for the calculation of the environmental impact. For *residential design*, the lifespan of the building is 75 years, whereas the lifespan of a *utilitarian design* is 50 years. In the case that there are multiple functions within the building, the user of the application should input the most unfavourable situation, which is the utilitarian situation because the life span of the utilitarian situation is shorter than the residential situation.

3.5.2 TYPE OF CALCULATION

The second decision the user must make is the type of calculation. This decision determines the way the calculation is formed, in which the user can choose between generic calculations or more specific calculations. Within this feature, the user chooses how specific the end-results have to be calculated.

In the early design stages, mainly the location of elements is specified, such as a wall, floor or roof. During the design process, more elements will be specified in terms of materialisation. Therefore, it was decided to have two types of calculations: the *generic* choice and *specific calculation* choice. Since the boundary of this research is set on the calculation of environmental impacts and costs of a design in early design stages and based upon the life-cycle phases mentioned in the literature research, the system boundaries for the calculations are done keeping the product phase (A1-A3) and the construction phase (A4-A5) in mind.

Considering the *generic* calculations, all the unspecified, generic building elements in the IFC-model are used to calculate the environmental and economic impact. The LCA and LCC data that is used to calculate this environmental and economic impact is based on the means of all the results for all available building materials per building element in the database. This data is extracted from the database based on the 4-digits NL/Sfb coding. By using the square meters related to the unspecified, generic building elements from the IFC-model and linking them to the mean values of the same building elements from the database, the environmental and economic impact can be calculated. For example, all the materials related to the exterior load-bearing wall type starting with assembly code “28.01” are used to determine the mean values of the environmental and economic impact. After the mean values are determined, these impacts will be combined with the square meters related to the exterior load-bearing wall type starting with assembly code “28.01” of the IFC-model. This combination creates generic results, based on the different building elements in a building design.

Because the design process evolves, decisions on which material type of load-bearing walls are used will be made. This classification of different material types means that a specific calculation can be made. Therefore, the user is enabled to choose the specific calculation type, which uses the data based on the 7-digits NL/Sfb coding. In other words, each specified building element with associated materials is calculated separately and is merged with the other specified building elements into a total specific impact score. For example, the data from both the external database and the IFC data related to the specified building elements with materials belonging to the assembly code “28.01” will be calculated separately and calculated into a total score for the environmental and economic impact of the specific calculations.

It must be stated that the specific calculation option will be a combination between generic and specific calculations, since, in the early design stages, not all the elements will be specified in detail. Therefore, the specific calculation is made based on the available specific element data and the remaining generic element data. In other words, the LOD of building elements in a design that the application can handle lies within the range of LOD 200 and LOD 300. This division of results will become more evident through the results of the case study, mentioned in the next chapter.

3.5.3 TYPE OF BUILDING SYSTEM

After the selection of calculation type the third and final choice the user must make is crucial for the provision of feedback by the application, since this decision will be used to generate potential improvements of the design. This choice implies what type of building system the user wants to review. Based on the decision of calculation choice combined with the type of building system, the application is enabled to import different materials linked to this building type, such as walls consisting of brick materials or floors consisting of concrete materials. Given the limited timeframe, this decision will be based on three criteria. These three criteria will be used for the testing of the case study, which are:

1. Brick/concrete Contains brick and concrete walls, and concrete floors;
2. Wood Contains floors and walls consisting of wood-elements;
3. All Contains the brick/concrete and wood-elements.

Within the overall building system, within the list of criteria referred to as “All”, all materials that have been specified within the database are used for the calculations. In other words, all the square meters of all available building elements are linked to their environmental and economic impact scores. In the case another building system choice is made, the square meters belonging to all building elements that are not included in a building system will be used as *generic* square meters and added to the total *generic* square meters of all unspecified building elements. In other words, the sum of all the areas of unspecified building elements based on only the 4-digits following the NL/Sfb coding will be added to the areas of the specific building elements that are not included in the chosen building system. Table 6 shows an example to clarify this principle, based on the different assembly codes and the corresponding square meters. The table contains the exterior wall building elements starting with assembly code “28.01”. The generic exterior walls are categorised by the 4-digit assembly code “28.01”, and the specific exterior walls are categorised by the 7-digit assembly codes “28.01.028”, containing wooden materials, and “28.01.029”, consisting of brick material.

Table 6 - Example of changes through the election of a type of building system

	Material	Calculation	The selected type of building system		
			All	Brick/Concrete	Wood
28.01	Generic	Generic	10 m ²	25 m ²	30 m ²
28.01.028	Wood	Specific	15 m ²	0 m ² (goes to generic)	15 m ²
28.01.029	Brick	Specific	20 m ²	20 m ²	0 m ² (goes to generic)

As can be seen in the Table 6, if wooden building elements have been specified in a model, but the user decides that these walls should consist of bricks or concrete as a building material, the user can select the building system type *brick/concrete* in the application in order to ignore the specified wooden building elements. In other words, the application filters all the brick and concrete building elements in the database and only uses these materials to make calculations, both based on the generic as well as the specific IFC-data. The IFC-data related to the specified wooden exterior walls will be converted to generic data and added to the existing generic IFC-data related to the assembly code “28.01”. This conversion enables the application to provide feedback based on the user’s preferences on which material is the best to use for that building element.

3.5.4 PREFERENCE OF IMPACT IMPORTANCE OF THE DESIGN

The final influence the user has on the different features within the application is the preference of impact importance of the design. Since the application is designed to provide

feedback on the optimisation of building materials based on their environmental and economic impact, the user decides what level of importance the impact has on their decision. In other words, the user has to select a preference of impact in order to get the optimal solution. For example, if the user decides to have a design that considers the environmental impact, the designer makes the decision to prioritize environmental impact in the application. However, it should also be possible to make an equally balanced decision, where the impact on the environment is equally as significant as the economic impact.

3.6 THE OUTPUT

The data-output, as well as the communication and interpretation of the results, is the final step of the process. The output stage comprises the combined impact of both the LCA and LCC results, presented in the form of comprehensive graphical and plotted outputs. These different outputs will need to be presented in such form that users can make more informed decisions for their designs regarding building products. In order to structure the output generated by the application, a graphical user interface (GUI) was developed. According to the different interviewees, the GUI of LEICAS had to be designed in such a way that anyone could use it. Following this advice, the GUI of LEICAS is designed to be an easy to use and efficient application that in base supports the users to make quick calculations for the determination of the environmental impacts and costing of their design. LEICAS consists of five components: (i) File tab; (ii) Manual user input; (iii) Calculation results within the *Overview*-tab; (iv) Calculation results for the overall scores in sidebar; and (v) Feedback. An overview of these components can be seen in Figure 21.

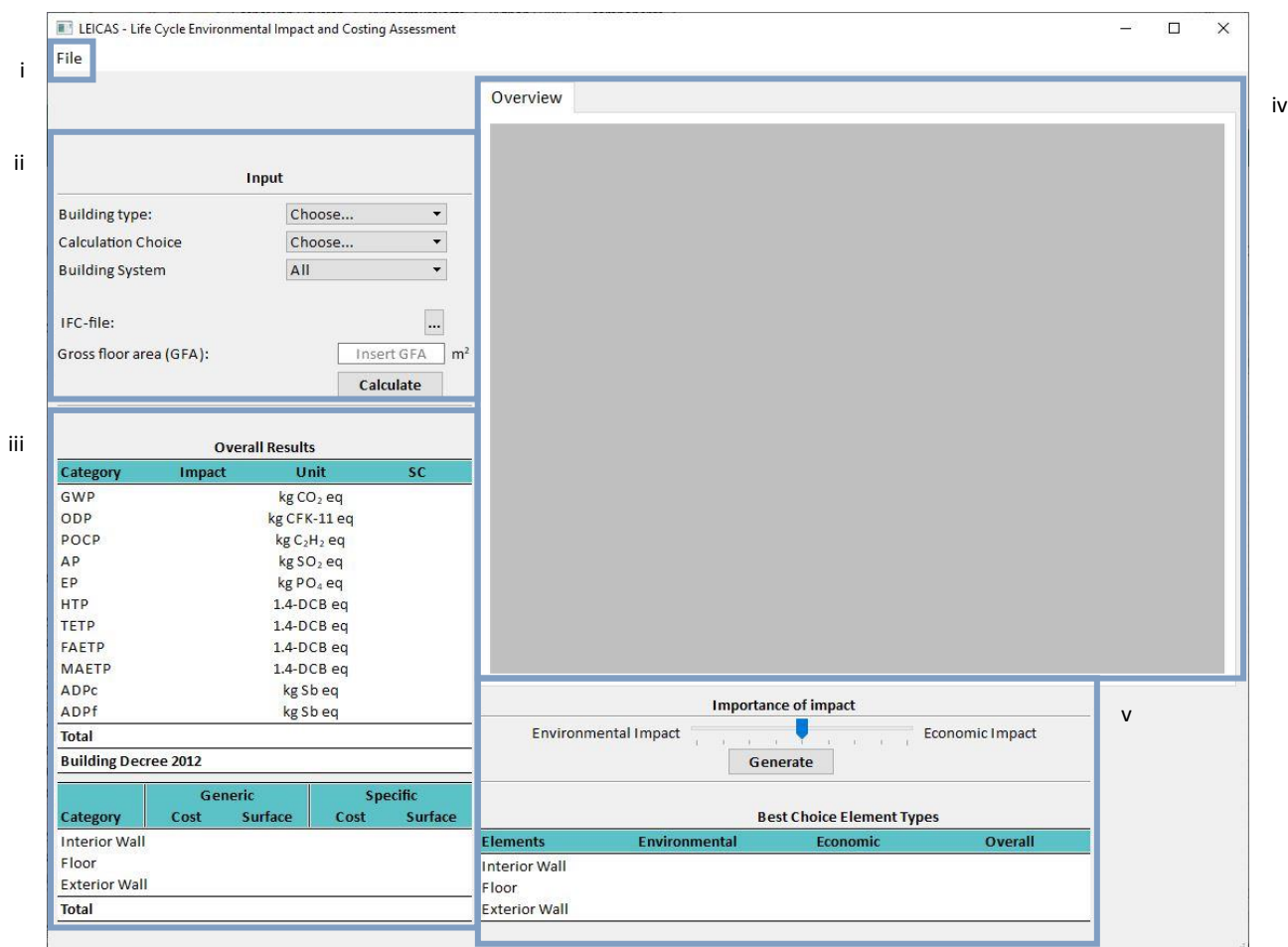


Figure 21 - Graphical User Interface of LEICAS

3.6.1 FILE TAB

The *File Tab* provides the user of the application with three options. The first option is the functionality of the *Open*, which enables the user to import an IFC-file into the application and exports a .xlsx-file of the required IFC-data for the calculations. The following functionality is *Save*, which enables the user to make an export based on the results in the *Overview* tab. The *Close* functionality closes the application for the user. These three options can also be entered by using different shortcuts on the keyboard based on a Windows-environment. For *opening* an IFC-file, the shortcut *Ctrl + O* can be used, whereas *closing* the application can be done by the shortcut *Esc*. If figures are presented in the application, using the *Ctrl + S* shortcut can be used to save the figures.

3.6.2 MANUAL USER INPUT

The manual user input provides the preferences and project-specific information of the user, which influences the LCA and LCC calculations, see section 3.5. Once the user sets the different variable features, the application can be run by either pressing the '*Calculate*' button or the '*Generate*' button. However, to enable the application to make calculations, an essential feature within the application is the import of an IFC-model. If the user has an IFC-model consisting of the elements mentioned in section 3.1.3, the model can be uploaded within the application. The user can upload an IFC-file by either clicking on the upload-button of the IFC-file or by opening the file within the "File tab". After the IFC-file is uploaded, the uploaded name of the file appears. This functionality lets the user know that an IFC-file is uploaded. Once the IFC-model is uploaded, the application will automatically export a .xlsx-file containing the following components: (i) Assembly Code; (ii) Element; (iii) Materialreference; (iv) Area; and (v) Width. An example of an .xlsx-file is shown in Figure 22.

Furthermore, if the IFC-model is uploaded, the LEICAS-application searches for the available *IfcSpaces* in the model and presents this within the application as gross floor area (GFA). If these are available, the sum of these *IfcSpaces* will form the GFA. This GFA is related to the maximum environmental impact score mentioned in Article 5.9 from the Dutch Building Decree 2012, which is expressed as one overall shadow price in euros per square meters of GFA per year. This article specifies that the shadow price of a newly built residential or utilitarian building with a gross floor area over 100 m² should be at most € 1.00 per m² of gross floor area per year.

If the *IfcSpaces* are not present in the IFC-model, the application uses the areas of the floors plus the areas of the exterior walls considering the lengths of the walls multiplied by the wall thicknesses. This way of calculating the GFA can be considered tricky since this calculation requires the IFC-model to be classified correctly. In other words, the building elements corresponding to the exterior walls and floors need to have the correct classification. If this correct classification is not the case, the GFA can be corrupt. Therefore, it has been decided that this feature can also be filled in manually. This manual input supports the user to make correct calculations based on their experience in what the true GFA is. As an extra benefit of the manual input, the user of the application can tweak the GFA which enables the opportunity to check if other GFAs can also be used in order to meet the requirements of the Dutch Building Decree regarding the maximum environmental impact score.

	A	B	C	D	E
1	Assembly Code	Element	Materialreference	Area	Width
2	22.03	Wall	-	10.00	100
3	22.03.001	Wall	-	5.00	200
4	22.03.016	Wall	-	100.00	100
5	22.03.024	Wall	-	20.00	100
6	22.04	Wall	-	0.00	200
7	22.05	Wall	-	0.00	200
8	23.01	Floor	-	120.00	250
9	23.01.003	Floor	-	25.00	200
10	23.01.072	Floor	-	25.00	200
11	28.03	Wall	-	200.00	200
12	28.01.003	Wall	-	25.00	200
13	28.01.028	Wall	-	55.00	200
14	28.01.029	Wall	-	30.00	200
15					

Figure 22 - Snippet of .xlsx-file based on the IFC-model

3.6.3 CALCULATION RESULTS

The results of the calculations are presented within the *overview* tab of the application as well as in the left sidebar. The difference between the presentations of the results is that in the left sidebar, the results are presented in a comprehensive overview which shows the overall results in a numerical display. Furthermore, within the left sidebar, the feature consisting of the acceptance or rejection of the required environmental impact score based on the Dutch Building Decree was added. This feature provides the user with the information whether or not the environmental performance of the building design matches the requirements set by Dutch policies. In the *overview* tab, bar charts are plotted to represent the environmental impact of materials per building element location on the left and the economic impact of materials per building element location on the right.

The content of the bar charts changes depending on the applied calculation options; generic and specific. When the calculation option is set on *generic*, the bar charts only present the results per material based on the overall generic square meters of the building elements. When the user chooses the *specific* calculation option, the bar charts will show a distinction in results of both the specific results as well as the generic results. These differences between the bar charts as well as the distinction of results within the *specific* calculation choice can be seen in Figure 23 and Figure 24.

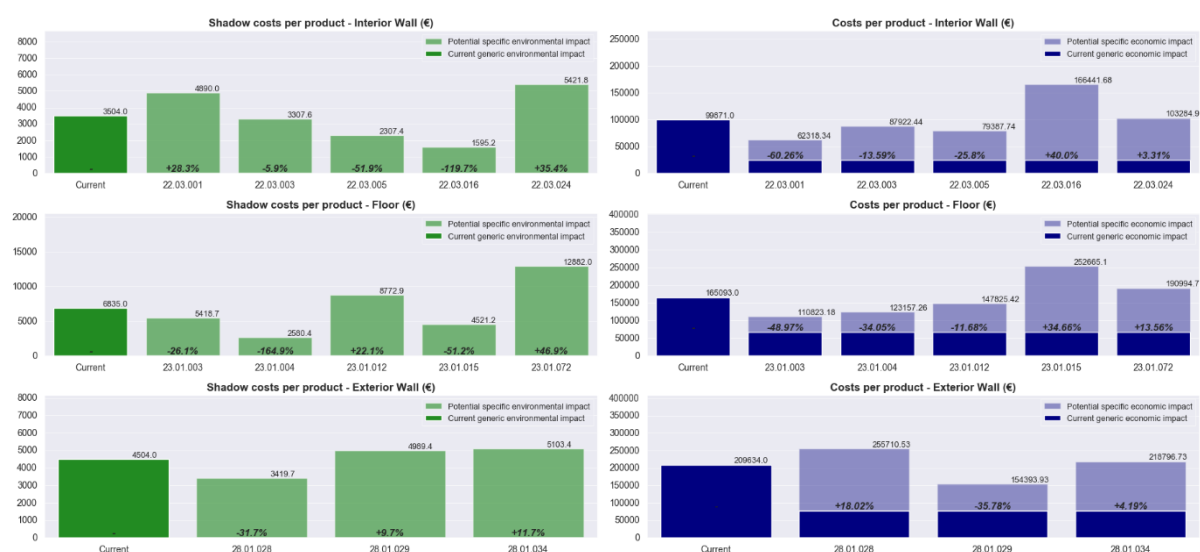


Figure 23 - Example of a bar chart based on the generic type of calculation

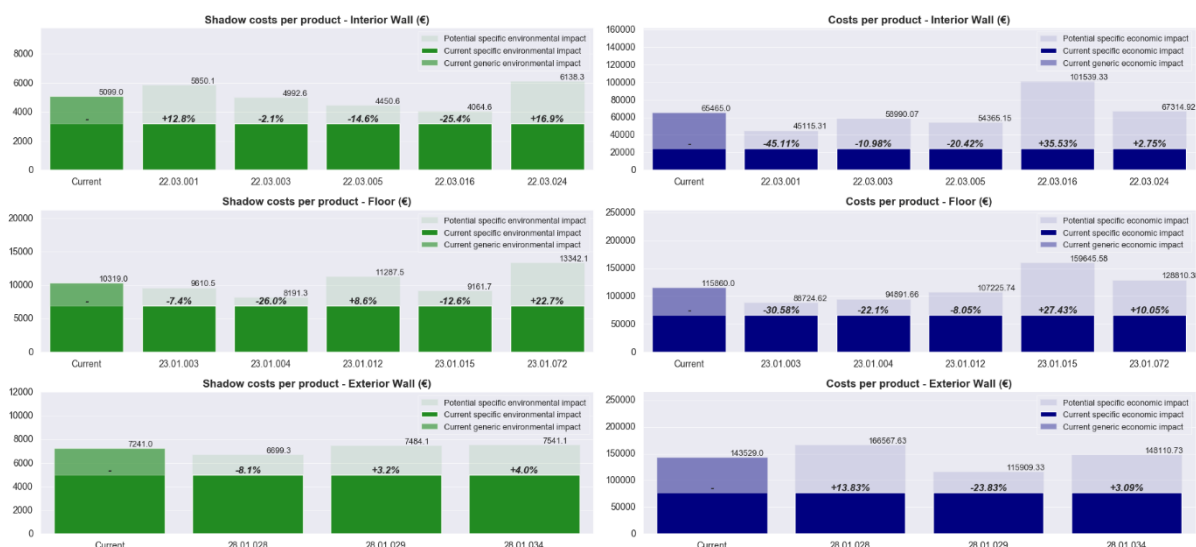


Figure 24 - Example of a bar chart based on the specific type of calculation

Another feature within the bar charts, is that of the percentages of impact. These percentages are based on the increase or decrease of the impact of the materials compared to the generic situation. In other words, the application provides the user with information that can be used to decide which materials are optimal to use in order to optimise the design. Besides, in each of the bar charts, the bar on the far left always presents the current situation of the calculations. The remaining bar charts present the different materials and their environmental and economic impacts.

3.7 THE FEEDBACK

The main feature of the application is *Feedback*. This feature will provide the user knowledge in how to optimise the design based on the user's preferences and based on the unclassified components of the design. Considering the decisions made in the *Manual User Input*, the application will provide an overview of materials that should be chosen based on: (i) the lowest environmental impact; (ii) the lowest economic impact; and (iii) the best solution considering the environmental as well as the economic impact. The application generates the lowest impact for both the environmental- as well as the economic impact of the materials based on the available materials and building system choice. In addition, on a scale from one to ten, the user can make a decision on how environmentally or economically considerate the design should be, as can be seen in Figure 21 (v). For example, if a user wants a design based on a more economical preference, the user must drag the scroller in the application to the side of the economic impact so that the importance of the economic impact is the preferred indicator. Based on this economic preference, the application will show the best solution with regards to the preferences of the user related to the environmental as well as the economic impact.

A notion must be made on this feature. Since the design in the early design stages is not always fully elaborated, the user still needs to consider which building systems should be chosen. Therefore, in order for the application to provide feedback, the user should provide all the available data. Furthermore, based on the decisions with regards to the building system, if a user chooses to work with a building system, it is highly recommended to work with a *specific* calculation method. This specific calculation method is recommended since when feedback is generated, a proposition of the most optimal solution is given based on different available

materials. In the case that a building system is specified while the calculation choice is set on *generic*, the feedback that is provided will only present generic results. These results will then only be a combination of all the mean values of the materials. Therefore, to get feedback on how to classify the unclassified components in order to optimise the building design, the user needs to set the calculation choice to *specific*.

3.8 FIELD OF IMPROVEMENTS

The LEICAS application automatically obtains the BoQ of all building elements that are present in an IFC-model and links this data to an LCA and LCC database of building products and materials. By linking the BIM-data with the environmental and economic database, the embodied environmental and economic impacts of design are calculated. Based on the environmental and economic results, the application provides an overview of materials and products that could be used to optimise the uncategorised building elements in a design.

The application provides an initial step into the implementation of LCA and LCC in the AEC-industry by linking the environmental and economic data to BIM objects and provide feedback in the optimisation of a design. However, there are still drawbacks within the functionality of the application and the usage of LCA and LCC as methods that need to be addressed in order to improve future versions of the application. These drawbacks have been discussed during an overall meeting where multiple experts in the AEC-industry provided live feedback on the functionality of the application. This overall meeting consisted of a live demonstration on how the application works, from the functionality of the *input* to the interpretation of the *feedback*. Afterwards, a Q/A session was held to list the different drawbacks. These drawbacks were then clearly reported and are further explained in Table 7.

Table 7 - Field of improvements

Improvements	Explanation
Input	The IFC-data that is extracted from the IFC-models is only tested using BIM-software packages Revit and Sketchup. Other BIM-software packages that provide IFC-exports such as ArchiCad and Tekla should be researched.
	The IFC-models were researched based on wall and floor elements. However, within building design, there are more elements present, such as roofs and windows, requiring different IFC-classifications.
	The used database with the LCA and LCC data was made in a Microsoft Excel format. Furthermore, the export of the IFC-model was exported in Microsoft Excel format as well. Potentially, the data transfer used for the functionality of the application should be stored and processed in an automatic database.
	The categorisation of building elements in the IFC-models was based on the NL/Sfb coding scheme. The challenge in this coding scheme is that each participant of a project who must provide an IFC-model, needs to categorise their building elements following the NL/Sfb scheme. This categorisation implies that each participant is bound to a way of working. The experts have stated that the challenge lies within the implementation of the NL/Sfb coding scheme by all small subcontractors.
Processing results	In order to combine LCA and LCC in the early design stages, the only calculations that were made were based on the first phase of a building's life-cycle: the manufacturing and construction phase. It is advised to expand the databases and calculation methods to a broader perspective, so that LCA and LCC can also be calculated in later design stages.
	Only walls and floors were used as building elements. In a future version, more building elements are necessary to show the full potential of the application.
Output	There was no 3D-visualisation of the BIM-model available. A potential improvement of the application is a 3D-visualisation of the BIM-model, which also contains information on which building components were "uncategorised" and "categorised". This categorisation leads to an understanding of which LOD the design would be at the time of assessment.
	Not every expert had an insight into what the results of the environmental impact were. Although the different categories mention the GWP or ODP, for example, categorisations such as CO ₂ or SO ₄ emission would give experts more feeling for the valuation of environmental impacts of their design.
Feedback	The output and feedback of the results are now based on NL/Sfb coding. This output requires the user to thoroughly understand the different coding related to the content of its building element. By having a library of functionality that provides the different building elements, designers and users of the application can understand the results and the potential improvements of the designs more efficiently.
	An automated feedback generator, that enables the user to modify the IFC-model within the application in order to optimise the design, would be a must-have in future versions.

4 CASE STUDY - APARTMENT BUILDING

To demonstrate the potential of LEICAS, a 4-storey apartment building design will be evaluated. This chapter will follow the structure of ISO 14040 for the assessment of the environmental and economic impacts of the design (International Standard Organization, 1997). Exporting the apartment building as IFC will enable the option of extracting the data referring to the square metres and widths of the building components. By combining the IFC-data with the LCA-LCC data, the environmental and economic impacts can be calculated and analysed.

In order to make a comparison between the functionalities of the application, two situations will be tested based on the building calculation choice: (i) the *generic scenario*, which contains only 4-digit building-elements; and (ii) the *specific scenario*, which contains a variation of 4-digit and 7-digit building-elements. The scenarios, based on building type *residential*, will be tested on the three-building systems combined, which results in six variations.

The results with regards to the environmental impact of the improved *specific scenario* generated by the LEICAS-application will be used in order to validate these results by calculating the environmental impact in other applications that utilises the MPG-methodology. For the economic impact, the *specific scenario* results will be validated using the free-to-use *BouwkostenOnline*.

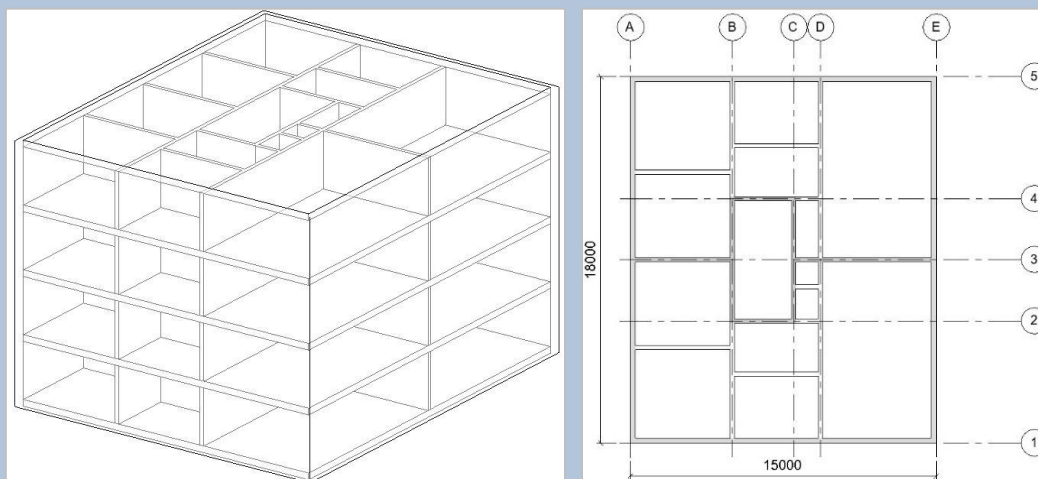


Figure 25 - Snippets from contouring of the apartment building

4.1 GOAL AND SCOPE DEFINITION

The main structure of the 4-storey apartment building consists of four exterior load-bearing walls and four floors. The apartment building consists of a centre core following standard grids. The GFA per floor is 270 square metre, which in total provides a GFA of 1080 square meters. The height per floor, which in this case is between the top of the floors, is 2.9 metres. These parameters are used for both scenarios, which will be explained in the next section.

For the *generic scenario*, the widths of the exterior load-bearing walls are 200 millimetres in width, whereas the inner walls are 100 millimetres in width. The floors are 280 millimetres in widths. In order to specify and improve the design in later stages and since the grids have been set out on an equal base, the following classification codes will specify the walls and floors of each situation:

- The interior load-bearing walls will be classified with the assembly code “22.03”, containing 856.74 square meters;
- Four floors will be classified with the assembly code “23.01”, containing 1284.8 square meters;
- Four exterior load-bearing walls will be classified with the assembly code “28.01”, containing 785.4 square meters.

In order to show the functionality of the application, a scenario is sketched where some materials of the building elements have already been chosen. For the *specific scenario*, the widths of the walls and floors will be the same as the *generic scenario*. However, the classifications of the different building elements will be specified according to a variety of walls and floors with classification codes:

- Twenty-four interior walls will be classified with the assembly code “22.03.001”, containing 303.92 square meters;
- Forty-four interior walls will be classified with the assembly code “22.03”, containing 552.82 square meters;
- Two floors will be classified with the assembly code “23.01.072”, containing 513.92 square meters;
- Two floors will be classified with the assembly code “23.01”, containing 770.88 square meters;
- Two exterior walls will be classified with the assembly code “28.01.029”, containing 392.70 square meters;
- Two exterior walls will be classified with the assembly code “28.01”, containing 372.70 square meters.

4.2 INVENTORY ANALYSIS

For the inventory analysis, the data required for the calculation of the environmental and economic impact are provided. The LEICAS-application utilises the precalculated data from the LCA and LCC database. Therefore, it is not necessary to search for additional LCA and LCC data since these data is already provided in the database. Other data that are still required for the calculation is the building type of design. Considering the MPG-methodology, a distinction is made between a utilitarian and residential building. Since the scenario is set up for apartments, the design will require calculations based on the parameter of residents.

4.3 IMPACT ASSESSMENT

For the impact assessment stage, the LEICAS-application will be used in order to identify the building components extracted from the IFC-model. The data extracted from the IFC-model will be used to calculate the environmental and economic impacts of the design in combination with the users' preferences. The results of all the different combinations of both the *generic scenario* as well as the *specific scenario* are presented in Appendix V. However, to get a feeling of how the results are presented, Figure 26 shows the results of the *specific scenario*. It can be stated that in terms of environmental impact, the highest contributors to the total shadow costs of this design are the global warming potential (GWP), acidification potential (AP), and the human toxicity potential (HTP). The lowest contributors, which are the abiotic depletion of composites (ADPc) and the stratospheric ozone depletion potential (ODP), are significantly lower than the rest of the contributors.

Altogether, the total shadow cost of the design is of €18,163.08. As can be seen in the bar charts in Figure 26 related to the shadow costs per product, the best solution to optimise the design by specifying the unspecified "generic" building elements will influence the shadow cost as well. Especially with regards to the assignments of materials to the floors, since in overall the environmental impacts of the materials of the floors form the highest contribution to the shadow costs. This high contribution by floors can be explained by looking at the unspecified "generic" square meters mentioned within the costing table in the left corner. This table shows that within the 3D-model 771 square meters of floors are still unassigned to a material, whereas the interior walls have 553 square meters and the exterior walls have 393 square meters still unassigned. The conclusion that can be drafted from the output of the application with regards to the environmental impact is the score related to the Dutch Building Decree 2012. The calculated shadow cost of 0.22 per square meter of GFA per year is lower than the required threshold of 1.0 per square meter GFA per year. Therefore, based on the regularities of the Dutch Building Decree, the current building design is accepted based on the maximum accepted environmental impacts.

With regards to the economic impact of the design, it can be stated that in general, the interior walls have the lowest contribution to the total economic impact of the design. As have been mentioned earlier, there are still building elements that do not have a material assigned. These building elements can, therefore, be optimised by using the table of "*Best Choice Element Types*". By using this table, the user is enabled to decide based on its preferences, such as the lowest economic impact. If this choice was made by the designer, the most optimal solution for the assignment of unclassified "generic" building elements are interior wall-type "22.03.001", floor-type "23.01.003", and "28.01.029" for the exterior wall-type.

Overall, based on the means of the generic costs, the total costs for the unspecified building elements will be € 189,604.82 and for the specified building elements € 196,473.24. Although the generic costs are lower than the specific costs, it must be mentioned that the generic costs will be formed based on the mean of all costs of that building element. Although this generalisation provides a preliminary estimation of the costs per building element type, the estimation must be considered as a rough estimation and should not be used as a well-grounded calculation. If a user wants to make more precise calculations, it is highly recommended to assign materials to the "generic" building elements in order to get a better base for further cost calculations.



Figure 26 - Snippet with results of the specific scenario based on building system "All"

As have been mentioned earlier, the results of the different choices that can be made by a user of the application are presented in Appendix V. However, Table 8 provides an overview of the results of the calculations of the LEICAS-application related to the specific calculations of the *specific scenario*. Based on these results, the unclassified building components are given a classification in order to optimise that building component.

Table 8 - Results of the LEICAS-application, optimising the building component of the *specific scenario*

	Different Impacts	Building system choice		
		All components	Bricks/concrete	Wood
Interior walls	Lowest Economic Impact	22.03.016	22.03.016	22.03.001
	Lowest Combined Impact	22.03.001	22.03.005	22.03.001
	Lowest Combined Impact	22.03.005	22.03.005	22.03.001
Floors	Lowest Environmental Impact	23.01.004	23.01.003	23.01.004
	Lowest Economic Impact	23.01.003	23.01.003	23.01.004
	Lowest Combined Impact	23.01.004	23.01.003	23.01.004
Exterior walls	Lowest Environmental Impact	28.01.028	28.01.034	28.01.028
	Lowest Economic Impact	28.01.029	28.01.029	28.01.028
	Lowest Combined Impact	28.01.034	28.01.034	28.01.028

4.4 INTERPRETATION STAGE

The results generated by the LEICAS-application will be validated through the optimisation of the *generic scenario*, based on the *building system choice* "all components". For the validation of the optimal solution, the different *unclassified components* are changed to their optimal solution with the *lowest combined impact*. With this new information, the data will be validated through different applications for both the environmental as well as the economic impact. In other words, the changes mentioned in Table 9 are made.

Table 9 - Changing the *generic scenario* building assembly codes to the optimised assembly codes

Category	Generic data (current)	Optimised situation (new)
Interior Walls	22.03	22.03.005
Floors	23.01	23.01.004
Exterior Walls	28.01	28.01.034

4.4.1 ENVIRONMENTAL IMPACT

For the validation of the environmental impact of the optimal design, calculations based on the new data will be conducted by using MPGCalc as well as the MPG-ENVIE. Both applications integrate the products and materials available in the NMD, which are also as a base for the database of the LEICAS application. The difference between the MPGCalc and the MPG-ENVIE is that the MPGCalc requires users to select the products based on the quantities of the design. These quantities are added manually by the user. MPG-ENVIE is an application which uses the quantities of the BIM-model in order to calculate the environmental impacts. Table 10 shows the optimised *generic scenario* results of the environmental impacts within the three applications.

Table 10 - Comparing the optimised *generic scenario* results of the environmental impacts

Impact category	Environmental impact (kg-eq)			Shadow cost (€)			
	LEICAS	MPGCalc	MPG-ENVIE	LEICAS	MPGCalc	MPG-ENVIE	Difference
GWP	97,063.96	93,431.70	97,064.0	€ 4,853.20	€ 4,672.000	€ 4,853.20	3.9%
ODP	0.01	0.0078	0.01	€ 0.24	€ 0.23	€ 0.24	4.3%
POCP	63.66	62.2	63.7	€ 127.33	€ 124	€ 127.33	2.7%
AP	411.08	387.3	411.1	€ 1,644.31	€ 1,549.000	€ 1,644.31	6.2%
EP	93.5	90.6	93.5	€ 841.53	€ 815	€ 841.53	3.3%
HTP	29,981.16	28,263.30	29,981.2	€ 2,698.30	€ 2,544.000	€ 2,698.30	6.1%
TETP	366.01	341.3	366.0	€ 21.96	€ 20	€ 21.96	9.8%
FAETP	767.54	760.7	767.5	€ 23.03	€ 23	€ 23.03	0.1%
MAETP	3,580,053.27	3,567,736.70	3,580,053.3	€ 358.01	€ 357	€ 358.01	0.3%
ADPc	0.2	0.2	0.2	€ 0.03	€ 0.03	€ 0.03	0.0%
ADPf	483.0	470.6	483.0	€ 77.28	€ 75	€ 77.28	3.0%
Total	-	-	-	€ 10,645.22	€ 10,179.26	€ 10,645.22	4.6%
Shadow cost (SC/m ²)	-	-	-	€ 0.13	€ 0.13	€ 0.13	0.0%

By examining the data, it can be stated that the results in both the LEICAS application as well as within the MPG-ENVIE are quite similar. This similarity can be explained by the fact that both applications use the same type of database and use the same method for calculating the environmental impact. However, slight differences can occur in further stages where the designs have higher LODs. These differences occur since the LEICAS-application can make environmental calculations of any building element thickness, whereas the MPG-ENVIE can only calculate environmental impacts of building elements containing fixed element thicknesses, resulting in different factors that influence the environmental impacts.

However, based on the generic situation, the differences between the MPGCalc and the MPG-ENVIE or the LEICAS application can be considered significant, especially with regards to the results from the GWP, AP and HTP as can be seen in Table 10. In theory, the utilised database from MPGCalc is the same as the ones used in the MPG-ENVIE and the LEICAS application. However, MPGCalc does not provide any underlying explanation of how the database was used and how the calculations were made based on the information of the database. It is, therefore, hard to assess how the differences between the different applications occur. A logical explanation could be that the correctional factors used in both the MPG-ENVIE as well as in the LEICAS application are stricter than the factors within the MPGCalc.

Another explanation could be related to the utilised data in the applications. Although the same database is used in the three applications, it seems reasonable to assume that the

database is updated and that a different version of the database is used for the calculations in the applications. Altogether, the percentual differences between the three applications become less apparent when the individual shadow costs are put together. Eventually, the shadow price per square meter in the three applications is the same, € 0.13.

4.4.2 ECONOMIC IMPACT

For the validation of the economic impact of the optimal design, calculations based on the new data will be conducted by using *BouwkostenOnline* database. *BouwkostenOnline* is an application which has a wide range of data. The *same owner owns the BouwkostenOnline* as of the used LCC-database for the development of the LEICAS-application. However, the developers of the *BouwkostenOnline* application are different from the used LCC-database. The *BouwkostenOnline* application requires users to select the products corresponding to the design manually. The application will make calculations based on the input data from the user. Table 11 shows the optimised *generic scenario* results of the economic impacts.

Table 11 - Comparing the optimised *generic scenario* results of the economic impacts

Category	LEICAS	<i>BouwkostenOnline</i>	Difference
Interior Walls	€ 54,951.3	€ 50,093.59	8.84%
Floors	€ 71,589.06	€ 68,629.73	4.13%
Exterior Walls	€ 141,529.08	€ 183,147.43	-29.41%
Total	€ 268,069.44	€ 301,870.75	-12.61%

The results from both applications are different. The biggest difference that can be noted is the significant difference in cost of wall-types consisting of assembly code 28.01. The explanation of this difference in results occurs because each database uses different parameters and data to make cost-calculations. This difference has the effect that the total results of each building element will be different. In addition, the differences are also in line of expectation since the economic valuation of building materials fluctuate over time. Related to the application, the used databases for the cost-calculations in the applications could differ from each other in terms of used data. In other words, the data used in the *BouwkostenOnline* application can be deprecated and not be conform to current prices of materials.

4.4.3 SENSITIVITY CHECK OF THE ENVIRONMENTAL AND ECONOMIC RESULTS

In order to test the sensitivity of the results, besides the change of the abovementioned assembly codes, the building elements will be changed. The building elements suggested for the sensitivity check are based on a concrete structure with environmental interior walls. To be more specific, the interior walls will contain the assembly code 22.03.001, the floors will contain the assembly code 23.01.072, and the exterior walls consist of assembly code 28.01.034. Based on this sensitivity analysis, the results with regards to the environmental impact are shown in Table 12, and the economic impact is shown in Table 13.

As can be seen in Table 12, in general, the differences between the different environmental impact categories of the different applications can be considered not significant compared to each other. In other words, the differences in environmental impacts, considering the major contributors to the total environmental impact score do not have a significant impact on the result. The only “significant” differences are highlighted in red. These differences do not have a significant contribution to the total score; they form less than one percent of the total shadow price. Furthermore, the final shadow price per square meter is equal in each

application. However, in comparison to the environmental impact scores of the optimised *generic scenario*, instead of having a positive result of differences in the optimised *generic scenario*, the environmental impact scores in the sensitivity analysis scenario, in general, are more negative than positive. Furthermore, the major contributors to the total impact score seem to have levelled out to meet each other among the different applications. Altogether, differences in the impact categories still occur. Therefore, the formerly mentioned theory is encouraged that either the utilised data is different or both the utilised data as well as the calculation rules within the different applications are different. Future work must investigate these differences and how these differences can occur.

Table 12 - Sensitivity analysis comparison of the environmental impacts in different applications

Impact category	Environmental impact (kg-eq)			Shadow cost (€)			
	LEICAS	MPGCalc	MPG-ENVIE	LEICAS	MPGCalc	MPG-ENVIE	Difference
GWP	221,194.22	221,947.5	221,194.2	€ 11,059.71	€ 11,097	€ 11059.71	- 0.3%
ODP	0.01	0.0129	0.01	€ 0.37	€ 0.39	€ 0.37	- 5.1%
POCP	122.18	123.8	122.2	€ 244.36	€ 248	€ 244.36	- 1.5%
AP	740.94	752.0	740.9	€ 2,963.75	€ 3,008	€ 2,963.75	- 1.5%
EP	157.4	160.4	157.4	€ 1,416.62	€ 1,443	€ 1,416.62	- 1.8%
HTP	107,586.26	107,165.6	107,586.3	€ 9,682.76	€ 9,645	€ 9,682.76	0.4%
TETP	2110.39	2,078.3	2110.4	€ 126.62	€ 125	€ 126.62	1.3%
FAETP	1093.97	1,131.5	1094.0	€ 32.82	€ 34	€ 32.82	- 3.5%
MAETP	4,272,558.24	4,438,787.0	4,272,558.2	€ 427.26	€ 444	€ 427.26	- 3.8%
ADPc	0.42	0.4	0.4	€ 0.07	€ 0.06	€ 0.07	16.7%
ADPf	1002.54	1,020.1	1002.5	€ 160.41	€ 163	€ 160.41	- 1.6%
Total	-	-	-	€ 26,114.75	€ 26,207	€ 26,114.75	- 0.4%
Shadow price (SC/m ²)	-	-	-	€ 0.32	€ 0.32	€ 0.32	0.0%

As can be seen in Table 13, the differences between the different economic impacts per building element with regards to the sensitivity analysis still occur. However, it must be stated that this difference can be explained by looking at the highest contributor to the differences. The same exterior wall-type was used in both situations, which results in the same significant difference in exterior walls and total scores between the different applications. On the other hand, looking at the results of the interior walls and floors, the differences between the applications seem to be similar. This similarity means that the data in the databases of both applications are almost similar. Therefore, it can be assumed that the utilised data in the databases, based on the changed materials in the sensitivity analysis, are comparable with each other. Overall, it can therefore be concluded that only due to price fluctuations and up to date databases, different results within the applications can occur.

Table 13 - Sensitivity analysis comparison of the economic impacts in different applications

Category	LEICAS	BouwkostenOnline	Difference
Interior Walls	€ 37,987.85	€ 37,996.42	- 0.02%
Floors	€ 155,460.80	€ 155,447.95	+ 0.01%
Exterior Walls	€ 141,529.08	€ 183,147.43	- 29.41%
Total	€ 334,977.73	€ 376,591.80	- 12.42%

5 CONCLUSION

The conclusion forms the final chapter of the thesis. Within this final chapter, the research question will be answered by discussing the different sub-questions. Afterwards, the content of this thesis will be discussed. Finally, recommendations for future research and potential improvements of this thesis will be given.

5.1 THE RESEARCH QUESTION

This thesis explains the interest and opportunities of integrating LCT, especially LCA and LCC, with BIM during the early stages of a design by providing answers on how to integrate LCA and LCC within a BIM-based environment. Furthermore, integrating LCA and LCC within BIM can enable an automated environmental, and economic impact assessment of the whole life embodied impacts of building elements during the early design stages. Therefore, the primary objective of this research is to prove that a BIM-based assessment application can guide the decision-making process of designers by generating an automated environmental and economic impact assessment and providing guidance in the form of correct response feedback to the designers. The following research question was answered:

How can the integration of *Life-Cycle Assessment* and *Life-Cycle Costing* with *Building Information Modelling* support designers in decision-making by enabling an automated *Environmental and Economic Impact Assessment* during the early design stages?

To stimulate the application of LCA and LCC in the early design stages, a BIM-based application was developed to allow designers to quickly comprehend the embodied environmental and economic impacts of their designs. A case study of an apartment building was used to describe how the application works with regards to the integration of LCA and LCC with BIM. An answer to the main research question will be formulated based on the answers of the five sub-questions. These sub-questions are discussed next.

5.1.1 TO WHAT EXTENT ARE LCA AND LCC CURRENTLY APPLIED WITHIN THE AEC-INDUSTRY?

The general methodological framework of LCA can be directly applied to buildings. However, currently, there is not a general accepted methodological framework of LCC available. Furthermore, the application of both the LCA as well as the LCC approach in the AEC-industry remains a challenging task due to the relatively long lifespan of buildings, a shorter lifespan of individual building elements and materials, the unique nature of buildings, varying distances to factories and the many changes in function and form a building may happen during its lifetime.

Sub-consequently, LCA and LCC studies in the AEC-industry cannot always be completed with the same level of detail as in other sectors. It has been acknowledged that there are four defined levels of LCA and LCC studies, being screening, simplified, complete and streamlined studies. Overall, it can be stated that the levels of LCA and LCC studies differ in terms of data representativeness, completeness of the assessment and documentation and communication (van Gemert, 2019). Screening and simplified LCAs are particularly useful in the early design stages to identify environmental hotspots, but for LCC, it can be quite challenging to implement the levels “screening” and “simplified”. Therefore, to combine LCA and LCC in the early design stages, the streamlined approach is recommended to use. According to Santos et al. (2019), in the case that quantitative data of the system is not available, which is the case in early design stages, the streamlined study is recommended for the assessment of different products with regards to their life-cycle. Although all these studies are mainly carried out within the generic framework of the ISO 14040 series of standards, related to LCA, they differ in terms of the applied functional unit, used inventory data and impact assessment method, studied impact categories and considered life-cycle phases or coverage of processes.

Furthermore, the development of tools has become a particularly important area of application to implement LCA and LCC in the AEC-industry. A challenge in utilising LCA and LCC by the AEC-industry lies within the complexity of LCA and LCC tools. Current assessment tools are quite complicated in terms of utilisation by designers or other parties without professional knowledge. In general, the tools consisting of the LCA and LCC calculation methods have been developed for a post-construction assessment of a building by which an LCA-expert utilises the tools. Therefore, the results of the tools are mainly applied for scientific research purposes and thus limits the comprehension by non-LCA and LCC experts during the development of a design. In other words, generally, the results cannot be interpreted by non-LCA and LCC experts, which results in the lack of implementation of LCA and LCC in the AEC-industry. In addition, during the design process, the different parties involved in a project need to collaborate, which means they may have different interests in design. For example, designers pay more attention to the environmental impact of design, while project managers focus on the economic impact of design. Current tools cannot meet these different needs simultaneously.

5.1.2 IN WHAT WAY CAN BIM BE UTILISED TO SUPPORT DESIGNERS IN MAKING ENVIRONMENTAL AND ECONOMIC DECISIONS BASED ON THE LCA AND LCC METHODOLOGY?

Generally, BIM can be considered as a process of generating, extracting and on some level even re-using building information during the lifespan of a building using models. Since BIM-models consist of parametric objects that are representing building elements, BIM-models can become useful for calculating the environmental and economic impacts of a building. These calculations can be made since the parametric objects contain information that can be linked to environmental and economic datasets. These so-called BoQs contain information such as identification, dimensions and types of elements.

Furthermore, BIM is an excellent tool to reduce manual data input and can provide a communicational bridge between multiple project participants. Essentially, BIM integrates the information on building elements within a BIM model with the assessment of environmental and economic impacts. The integration of BIM within the LCA and LCC approaches can, therefore, be a way to obtain fast and effective results for the determination of the environmental performance and the economic impact of a building. The facilitation of BIM establishes the application of LCA and LCC approaches in early design practice.

5.1.3 WHICH DIFFERENT FEEDBACK MECHANISMS INFLUENCE THE DECISION-MAKING OF THE ENVIRONMENTAL AND ECONOMIC IMPACT OF A BUILDING?

The applications that are on the market are based on providing information on the scores of the environmental and economic impacts of utilised materials in a building. In other words, the current applications only use simple verification feedback, which does not give any useful feedback on how to optimise designs. On the other hand, other applications in the AEC-industry that have integrated the LCA and LCC methodology within a BIM-based environment are using the try-again feedback approach. Although this approach provides information in the necessity of changing the design and sometimes even provide the option to adapt the design within the application itself, it does not present the designer any feedback into how the design should be improved.

Therefore, a useful feedback mechanism that should be implemented and utilised more within the LCA-LCC practice is the use of correct response feedback. This feedback mechanism informs the designer what the *correct response* is, which in the case of LCA and LCC in the AEC-industry implies that the *response* will be feedback on how to optimise a design based on the environmental and economic results for example. By providing such an approach, the user can make decisions based on the system's database and gets insight into how to optimise the building design.

5.1.4 WHAT ARE THE CONSTRAINTS OF A BIM-BASED ENVIRONMENTAL AND ECONOMIC IMPACT ASSESSMENT APPLICATION?

A constraint that was found in the literature and mainly not related to the functionality of the application is the reluctance of performing LCA and LCC studies by the AEC-industry. Although environmental life-cycle thinking is gaining interest in the AEC-industry, decision-making is mainly based on economic principles. Furthermore, current BIM-based environmental and economic impact assessment applications are ambiguous, meaning that only experts in the field of either environmental or economic advisement can use the applications. Besides, these applications mainly present their results and do not provide understandable feedback to the users. More applications such as the presented application in this thesis are required to stimulate the utilisation of environmental and economic impact assessment applications in the AEC-industry.

Another constraint that was found during the research was the cost-effectiveness of LCC in the short or long-term, which is a concern of using LCC as a method for economic impact assessment. The current cost of materials differs over time. Even on short term price fluctuation occurs frequently. By using LCC-methods such as the NPV-method, these price fluctuations can mean that at the start of a project, the design of a building is feasible in terms of cost, but at the end of a project, it could be the opposite. It is therefore vital to have a live, up to date LCC database that is used to calculate the economic impacts of a building. This up-to-date database has as advantage that during the process, economic decisions can be considered in order to get the most cost-effective design at any moment of the design. Of course, it must be noted that at a certain point of the design process, the design is finished, and only minor adjustments must be made in the design. Although in general LCA-databases consist of more static data than LCC-databases, by not updating LCA or LCC databases on regular basis inconsistencies and discrepancies can occur. As can be seen in the case-study, inconsistencies among different applications from both LCA as well as LCC perspective emerged. Altogether, it is therefore highly recommended to have a live economic database that changes on a highly regular database such as once a week or perhaps even more. With regards to the environmental LCA database, it is best to have a live database as well. However, this data does not need to be kept updated weekly; a monthly import of the data should be enough.

5.1.5 TO WHAT EXTENT IS DECISION-MAKING IN EARLY DESIGN STAGES ENHANCED BY APPLYING A BIM-BASED ENVIRONMENTAL AND ECONOMIC IMPACT ASSESSMENT APPLICATION?

Decision-making in the early design stages is enhanced using the LEICAS-application since it provides information about what the environmental and economic impacts are of a particular design. By testing the LEICAS-application in terms of performing calculations based on a BoQ of an IFC-model, it can be stated that the application is suitable for the assessment of

embodied environmental and economic impacts of an early stage schematic design and supports the users of the app in optimising the building design. The results were presented in an intuitive GUI which provides the presentation of the results based on a comprehensive underlying framework of interlinked python scripts.

The LEICAS-application provides an initial step into the implementation and integration of LCA and LCC within the AEC-industry during the early design stages; the results of the application are following the ad-hoc streamlined LCA and LCC-study approach. Such integration means that both the economic as well as the environmental results are comprehensive enough to be used in the early design stages.

5.2 DISCUSSION

It can be stated that this research provides a certain insight into the establishment of linking BIM with the LCA and LCC approaches in order to achieve environmental and economic considered decision-making in the early design stages. Furthermore, the conducted research shows a way of establishing a BIM-LCA-LCC integration within the AEC-industry. Since this research is an initial step into this integration, several points of discussion can be mentioned with regards to aspects such as the applied methodology as well as the output of the feedback.

5.2.1 CHOSEN APPROACH FOR BIM-LCA-LCC INTEGRATION

As the literature describes, there are three approaches to link LCA with LCC. However, only two of these approaches uses BIM as the foundation for the calculations related to the LCC and LCC. Within this research, the used approach used IFC-models where the BoQ was used to form the base for the environmental and economic calculations. By using this approach, the BIM data is exported to an external format. This export forms a significant drawback of having an IFC-export, because each time the user wants to calculate the environmental and economic impact of the building design, the user is required to perform an IFC-export. This IFC-export requires predetermined settings that the user must set in order to incorporate the required data of an IFC-export for calculating the environmental and economic impact. This additional step increases the amount of time spent on exporting the BIM-models and thus, the possibility of human errors increases as well. Therefore, recent researches tend to propose BIM-LCA-LCC integration solutions based on feedback opportunities within a BIM-based environment (Bueno, Pereira, & Fabricio, 2018; Röck, Passer, Ramon, & Allacker, 2019; van Gemert, 2019).

5.2.2 UTILISED LCA AND LCC METHODS

With regards to the used LCA database, the NMD, a new version of this database will become available in early 2020. This new version means that the current version of the database needs to be updated. Furthermore, the calculation method that uses this database, the MPG, uses this new version of the database. Because the database was not updated at the time of writing this thesis, it was found not possible to implement the new version of the MPG for calculating the environmental impact. Although this new version of the environmental calculation method is supposed to be similar to the current version (Milieudatabase, 2019), when the new database is presented, the application should be updated according to the new regulations and LCA data. The NMD database is required for the Dutch calculation method MPG. However, the question is if this database can be considered the best database available for calculating environmental impact. For example, the Ecolnvent-database, used as the base for the NMD,

is an extensive database with raw materials and their environmental impacts. This database is the world's leading environmental database. Another distributor, NIBE, has a database that can be quite extensive and helped set up the NMD as well. Besides these databases, there are many other LCA databases. It is questionable which database is the best database. Therefore, the decision of the LCA database should be made based on the regularities of the country of the project. With regards to the LEICAS application, since the LCA calculation method was based on the MPG and the NMD-database was formed based on the Ecolnvent-database as well as the NIBE-database, the decision to use the NMD as LCA-database is grounded.

By looking at the LCC database, it must be stated that in general, companies within the AEC-industry are using their economic calculation software. The data used for the economic calculations should, therefore, be based on the company's database. This company-driven database enables the users of the application to have calculations based on their data.

5.3 RECOMMENDATION

Overall, the research was to prove the true potential of integrating BIM with the LCA and LCC approaches. Therefore, the application that was developed can be an initial step into this integration. Drawbacks of the application are already mentioned in paragraph 3.4. Besides these drawbacks, other recommendations can be given based on current developments in the field of BIM-integration with LCA and LCC. Therefore, several recommendations related to potential future developments to enhance the LEICAS-application's functionality are further discussed.

5.3.1 LIFE-CYCLE STUDIES

Within the time span of the research plus the availability of data for the economic calculations, it was not feasible to dig deeper into the potentials of integrating LCA and LCC within later design stages, such as the technical design stages consisting of LOD 400 and beyond. Although this thesis provides an initial step into the potential of calculating LCA and LCC in the early design stages, it can be stated that later design stages are also exciting to research since the LOD is higher in these stages and thus the results will be more specific.

It must be stated that an LCC study in the early design stages can only be done based on the streamlined study approach. A possibility for the implementation of LCC in later stages would be to look at datasets that consist of crucial information to perform LCC calculations based on the NPV-method. This NPV-method provides fair results with regards to the LCC approach but can only be performed when data of the building is available.

5.3.2 BUILDING ELEMENTS

Significant development of the application in order to get a better understanding of the true potential of the application lies within the building elements. Since the application was only based on walls and floors, the environmental and economic scores of a building can be considered incomplete. For example, when executing a complete LCA and LCC-study, it is of high importance that the building entails the embodied impact of all building elements such as the walls and floors but also the roofs, slabs and windows. A possibility for future research would, therefore, be to incorporate other building elements in the application as well as the database. At first, it is recommended to look at the building envelope only before looking at installations.

As already has been discussed, the application does not take different variations of materials in terms of widths into account. However, to support the idea of adding more material variations within the calculations, the application already contains the option to export “*material-references*” of an IFC-model in the BoQ. This addition means that when the building elements are summarised in a BoQ, all the different widths based on these material references can be categorised and identified by the application. Further practical research can optimise the different python-scripts calculating the different impact scores based on the material variations of a building design.

5.3.3 IFC IMPORTS

The LEICAS application is especially developed to import IFC-models that are generated by BIM-software Revit and SketchUp, which are two of the major BIM-software contributors for building design. However, there are many more IFC-export tools on the market. Therefore, future work should be done in the exploration of how other design applications such as ArchiCAD could be integrated into the workflow of this application.

5.3.4 IMPLEMENTING S-LCA TO COMPLETE LCSA

Within this research and with regards to the LCSA, the two main pillars were considered: LCA and LCC. Since LCA and LCC form the most researched content of LCSA, the integration of LCA and LCC within this research forms an initial step to implement LCSA as an approach within the AEC-industry. In addition, LCA and LCC are two quantifiable approaches within the LCSA, which makes it possible to integrate both approaches. However, to form the LCSA, a third pillar should have been taken into account as well. By implementing social LCA into the calculation method, a more considered design can be developed based on the LCSA-principles.

Although social LCA is hard to research, a first step of implementing social LCA in the AEC-industry is by looking for a way to integrate BIM with social LCA. A similar approach to this research can be used in which BIM forms the starting point to integrate LCA with LCC. On the other hand, since social LCA concerns aspects such as employment, politics and human health, it can be considered impossible to implement certain aspects. Therefore, future research should look at the different aspects of social LCA that can be related to a building and its functionalities. A good starting point is by looking at the different functionalities of a building such as room functionalities, the orientation of the building and the primary function of the building, and how to match these different aspects to the perspectives of people on building variation choices. By collecting quantitative data on the perspectives of people, more insight can be gained into the usefulness of BIM-techniques with regards to social LCA.

Furthermore, the proposed research direction of social LCA only states how to integrate BIM with social LCA. However, at this point, social LCA is a subordinate towards LCSA. As this research acknowledges the reluctance of implementing LCSA with regards to LCA and LCC in the AEC-industry, it should also be researched if and why there is a reluctance of implementation of social LCA within the AEC-industry. Besides, how can social LCA form the AEC-industry? And in what way is social LCA applicable for the AEC-industry, or is the social LCA only applicable for parts of the AEC-industry such as the construction industry? There are a lot of questions that can be answered towards the implementation of social LCA in the AEC-industry. Future research should look at the potentials and the threats of social LCA in the AEC-industry and how it should be used.

5.3.5 MOST OPTIMAL SOLUTION

The most optimal solution for further development of the application, following the current BIM-LCA-LCC integration approach, would be that the export of the results and feedback is performed as .bcf export. The BCF-file, the BIM collaboration format, allows the different project participants, who presumably use different BIM software, to collaborate on issues within a project. Based on the different user's preferences used as input in the application and after the application generates the results and feedback on potential improvements of the design, the application should be able to export this information into a .bcf export file. By providing a BCF-export containing information on how to improve the design based on the users' preference, both from environmental as well as an economic perspective, other parties are enabled to update the design within their preferred BIM software directly. This BCF-file simplifies the communication of design changes between the different project participants and smoothens the execution of updating the BIM-model since non-BIM related reports, such as PDFs, are not required.

6 REFERENCES

- ACADAMY. (2008). *Streamlined Life Cycle Assessment*. Airbus.
- Ahmed, I., & Tsavdaridis, K. (2018). Life Cycle Assessment (LCA) and Cost (LCC) Studies of Lightweight Composite Flooring Systems. *Journal of Building Engineering*(20). doi:<https://doi.org/10.1016/j.jobe.2018.09.013>
- Ajayi, S. O. (2015). Life cycle environmental performance of material specification: A BIM-enhanced comparative assessment. *International Journal of Sustainable Building Technology and Urban Development* (6), 14-24. doi:<https://doi.org/10.1080/2093761X.2015.1006708>
- Ajayi, S., Oyedele, L., Akinade, O., Bilal, M., Owolabi, H., & Alaka, H. (2014). Ineffectiveness of construction waste management strategies: Knowledge gap analysis. *Proceedings of the First International Conference of the CIB Middle East and North Africa*, 261-280. doi:<https://doi.org/10.13140/RG.2.1.4788.8480>
- Akanbi, L., Oyedele, L., Akinade, O., Ajayia, A., Delgado, M., Bilala, M., & Bello, S. (2018). Salvaging building materials in a circular economy: A BIM-based whole-life performance estimator. *Resources, Conservation and Recycling* (129), 175-186.
- Akbarnezhad, A., Ong, K., & Chandra, L. (2014). Economic and environmental assessment of deconstruction strategies using building information modeling. *Automation in Construction* (37)(37), 131-144. doi:<https://doi.org/10.1016/j.autcon.2013.10.017>
- Akinade, O., Oyedele, L., Ajayi, S., Bilal, M., Alaka, H., Owolabi, H., . . . Kadiri, K. (2017). Design for Deconstruction (DfD): Critical success factors for diverting end-of-life waste from landfills. *Waste Management* (60)(60), 3-13. doi:<https://doi.org/10.1016/j.wasman.2016.08.017>
- Akinade, O., Oyedele, L., Bilal, M., Ajayi, S., Owolabi, H., Alaka, H., & Bello, S. (2015). Waste minimisation through deconstruction: A BIM based Deconstructability Assessment Score (BIM-DAS). *Resources, Conservation and Recycling* (105), 167-176.
- Álvarez Cid-Fuentes, J., Álvarez, P., Amela, R., Ishii, K., Morizawa, R., & Badia, R. (2019). Efficient development of high performance data analytics in Python. *Future Generation Computer Systems*. doi:<https://doi.org/10.1016/j.future.2019.09.051>
- Alwan, Z., Jones, P., & Holgate, P. (2017). Strategic sustainable development in the UK construction industry, through the framework for strategic sustainable development, using Building Information Modelling. *Journal of Cleaner Production* (140), 349-358.
- Anand, C., & Amor, B. (2017). Recent developments, future challenges and new research directions in LCA of buildings: A critical review. *Renewable and Sustainable Energy Reviews*, 408-416.
- Antón, L., & Díaz, J. (2014). Integration of life cycle assessment in a BIM environment. *Procedia Engineering* (85)(85), 26-32. doi:<https://doi.org/10.1016/j.proeng.2014.10.525>
- Azapagic, A. (1996). *Environmental System Analysis: The Application of Linear Programming to Life Cycle Assessment*. Guildford: University of Surrey.
- Azapagic, A., & Clift, R. (1999). Life cycle assessment and multiobjective optimisation. *Journal of Cleaner Production*(7(2)), 135-143. doi:[https://doi.org/10.1016/S0959-6526\(98\)00051-1](https://doi.org/10.1016/S0959-6526(98)00051-1)
- Azhar, S., & Brown, J. (2009). BIM for Sustainability Analyses. *International Journal of Construction Education and Research*(5(4)), 276-292. doi:<https://doi.org/10.1080/15578770903355657>

- Azhar, S., Hein, M., & Sketo, B. (2008). *Building Information Modeling (BIM): Benefits, Risks and Challenges*. Auburn: Auburn University. Retrieved from <https://www.researchgate.net/publication/237569739>
- Azhar, S., Nadeem, A., Mok, J., & Leung, B. (2008). Building Information Modeling (BIM): A New Paradigm for Visual Interactive Modeling and Simulation for Construction Projects. *Conference: First International Conference on Construction in Developing Countries (ICCIDC-I)*, (pp. 435-446). Karachi.
- Baez, J., Coya, B., & Rebro, F. (2018). Props in Network Theory. *Theory and Applications of Categories*(33(25)), 727-783.
- Bank, L., McCarthy, M., Thompson, B., & Menassa, C. (2010). Integrating BIM with system dynamics as a decision-making framework for sustainable building design and operation. *Proceedings of the First International Conference on Sustainable Urbanization*. Hong Kong: ICSU.
- Barlish, K., & Sullivan, K. (2012). How to measure the benefits of BIM - A case study approach. *Automation of Construction*(24), 149-159. doi:<https://doi.org/10.1016/j.autcon.2012.02.008>
- Barros Telles do Carmo, B., Margni, M., & Baptiste, P. (2016). Social impacts profile of suppliers: a S-LCA approach. *IFAC-PapersOnLine*(49-2), 36-41. doi:<https://doi.org/10.1016/j.ifacol.2016.03.007>
- Basbagill, J., Flager, F., Lepech, M., & Fischer, M. (2013). Application of life cycle assessment to early stage building for reduced embodied environmental impacts. *Building and Environment*(60), 81–92. doi:<https://doi.org/10.1016/j.buildenv.2012.11.009>
- Bilal, M., Oyedele, L., Qadir, J., Munir, K., Akinade, O., Ajayi, S., & Owolabi, H. (2016). Analysis of critical features and evaluation of BIM software: towards a plug-in for construction waste minimization using big data. *International Journal of Sustainable Building Technology and Urban Development* (6), 211-228.
- BIMForum. (2019). *BIMForum.org*. Retrieved from LOD: <https://bimforum.org/lod/>
- Blanchard, B. (1978). *Design and Manage to Life Cycle Cost*. Dilithium Pr.
- Bouwen met Staal. (2019, June). *Determining the environmental performance of buildings*. Retrieved from Sustainable in Steel: https://www.sustainableinsteel.eu/p/515/environmental_performance.html
- Brander, M., Tipper, R., Hutchison, C., & Davis, G. (2009). *Consequential and Attributional Approaches to LCA: A Guide to Policy Makers with Specific Reference to Greenhouse Gas LCA of Biofuels*. Retrieved from https://www.researchgate.net/publication/237335622_Consequential_and_Attributional_Approaches_to_LCA_A_Guide_to_Policy_Makers_with_Specific_Reference_to_Greenhouse_Gas_LCA_of_Biofuels
- Bribián, I., Capilla, V., & Usón, A. (2011). Life cycle assessment of building materials: Comparative analysis of energy and environmental impacts and evaluation of the eco-efficiency improvement potential. *Building and Environment* (46), 1133-1140.
- Bryde, D., Broquetas, M., & Volm, J. (2013). The project benefits of Building Information Modelling (BIM). *International Journal of Project Management*(31), 971-980. doi:<https://doi.org/10.1016/j.ijproman.2012.12.001>
- Buchert, T., Neugebauer, S., Schenker, S., Lindow, K., & Stark, R. (2015). Multi-criteria decision making as a tool for sustainable product development – Benefits and obstacles. *12th Global Conference on Sustainable Manufacturing*. doi:<https://doi.org/10.1016/j.procir.2014.07.110>

- Budzinski, M., Sisca, M., & Thrän, D. (2019). Consequential LCA and LCC using linear programming: an illustrative example of biorefineries. *The International Journal of Life Cycle Assessment*(24(12)), 2191–2205. doi:<http://doi.org/10.1007/s11367-019-01650-6>
- Bueno, C., Pereira, L., & Fabricio, M. (2018). Life cycle assessment and environmental-based choices at the early design stages: an application using building information modelling. *Architectural Engineering and Design Management*(14(5)), 332-346. doi:<https://doi.org/10.1080/17452007.2018.1458593>
- Buonocore, E., Franzese, P., Mellino, S., Ulgiati, S., Vigliaand, S., Zucaro, A., . . . Panula-Ontto, J. (2009). *Energy and LCA evaluation of the present dynamics of investigated case studies*. Synergies in Multi-scale Inter-Linkages of Eco-social systems. Turku: Socioeconomic Sciences and Humanities.
- Buyle, M., Braet, J., & Audenaert, A. (2013). Life cycle assessment in the construction sector: A review. *Renewable and Sustainable Energy Reviews*(26), 379-388. doi:<https://doi.org/10.1016/j.rser.2013.05.001>
- Cabeza, L., Rincón, L., Vilariño, V., Pérez, G., & Castell, A. (2014). Life cycle assessment (LCA) and life cycle energy analysis (LCEA) of buildings and the building sector: A review. *Renewable and Sustainable Energy Reviews* (29), 394-416. doi:<https://doi.org/10.1016/j.rser.2013.08.037>
- Capaz, R., & Seabra, J. (2016). Chapter 12 - Life Cycle Assessment of Biojet Fuels. *Biofuels for Aviation - Feedstocks, Technology and Implementation*, 279-294. doi:<https://doi.org/10.1016/B978-0-12-804568-8.00012-3>
- Caspeele, R., Taerwe, L., & Frangopol, D. (2018). Life Cycle Analysis and Assessment in Civil Engineering. *Towards an Integrated Vision* (p. 564). Ghent: CRC Press.
- Chang, Y.-J., Neugebauer, S., Lehmann, A., Scheumann, R., & Finkbeiner, M. (2017). Life Cycle Sustainability Assessment Approaches for Manufacturing. In R. Stark, G. Seliger, & J. Bonvoisin, *Sustainable Manufacturing - Challenges, Solutions and Implementation Perspectives* (pp. 221-237). Springer.
- Chien, K., Wu, Z., & Huang, S. (2014). Identifying and assessing critical risk factors for BIM projects: Empirical study. *Automation in Construction*(45), 1-15. doi:<https://doi.org/10.1016/j.autcon.2014.04.012>
- CIRCE. (2013). *BIM, IPD, LCA and LCC during the design, construction and operation phases*. New Energy Efficient Demonstration for Buildings.
- Ciribini, A., Ventura, S., & Paneroni, M. (2016). Implementation of an interoperable process to optimise design and construction phases of a residential building: A BIM Pilot Project. *Automation in Construction*, 62-73. doi:<https://doi.org/10.1016/j.autcon.2016.03.005>
- Ciroth, A., & Franze, J. (2009). *Life Cycle Costing in SimaPro*. Berlin: GreenDeltaTC.
- Curran, M. (1993). Broad-based environmental life cycle assessment. *Environmental Science & Technology* (27), 430-436. doi:<https://doi.org/10.1021/es00040a001>
- Curran, M. (2006). *Life Cycle Assessment: Principles and Practice*. Cincinnati: National Risk Management Research Laboratory.
- Curran, M. (2013). Life Cycle Assessment: a review of the methodology and its application to sustainability. *Current Opinion in Chemical Engineering*(2), 273-277. doi:<https://doi.org/10.1016/j.coche.2013.02.002>
- Curran, M., & Todd, J. (1999). *Streamlined Life-cycle Assessment: A Final Report From the SETAC North American Streamlined LCA Workgroup*. Pensacola: Society of Environmental Toxicology and Chemistry. Retrieved from

- <https://pdfs.semanticscholar.org/8ca0/ac01b77b5f68a96df0de7b4d59cfc827b125.pdf>
- de Vries, B., Allameh, E., & Heidari Jozam, M. (2012). *Smart-BIM (Building Information Modeling)*. Eindhoven: Eindhoven University of Technology. doi:<https://doi.org/10.22260/ISARC2012/0001>
- Dempsey, J., Driscoll, M., & Swindell, L. (1993). Text-Based Feedback. In J. Dempsey, & G. Sales, *Interactive instruction and feedback* (pp. 21-54). Englewood Cliffs, N.J.: Educational Technology Publications. Retrieved from https://books.google.nl/books?hl=nl&lr=&id=ss3fz-5WC6gC&oi=fnd&pg=PA21&ots=uz3LvtH8P_&sig=Umt_cL2TouKIKPomVzKlmJKU_hM&redir_esc=y#v=onepage&q&f=false
- Díaz, J., & Antón, L. (2014). Sustainable Construction Approach through Integration of LCA and BIM Tools. *Computing in Civil and Building Engineering*, 283-290.
- Dimyadi, J., Spearpoint, M., & Amor, R. (2008). Sharing Building Information using the IFC Data Model for FDS Fire Simulation. *Fire Safety Science*. doi:<https://doi.org/10.3801/IAFSS.FSS.9-1329>
- Drejeris, R., & Kavolynas, A. (2014). Multi-Criteria Evaluation of Building Sustainability Behavior. *Procedia - Social and Behavioral Sciences* (110), 502-511.
- Eastman, C. M. (1976). General purpose building description systems. *Computer-Aided Design*(8), 17-26. doi:[https://doi.org/10.1016/0010-4485\(76\)90005-1](https://doi.org/10.1016/0010-4485(76)90005-1)
- Eastman, C., Teicholz, P., Sacks, R., Liston, & K. (2011). *BIM Handbook - A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*. Hoboken: John Wiley & Sons, Inc.
- Edelen, A., & Ingwersen, W. (2016). *Guidance on Data Quality Assessment for Life Cycle Inventory Data*. Cincinnati: Environmental Protection Agency. Retrieved from https://cfpub.epa.gov/si/si_public_record_report.cfm?Lab=NRMRL&dirEntryId=321834
- Ekvall, T., Azapagic, A., Finnveden, G., Ryderg, T., Weidema, B., & Zamagni, A. (2016). Attributional and consequential LCA in the ILCD handbook. *The International Journal of Life Cycle Assessment*(21-3), 293-296. Retrieved from <http://rdcu.be/mWtV>
- Ellen Macarthur Foundation. (2013). *Towards the Circular Economy, Opportunities for the consumer goods sector*. Ellen Macarthur Foundation.
- Ellen MacArthur Foundation. (2015). *Growth within: A Circular Economy vision for a competitive Europe*.
- ENSLIC. (2010). *Energy Saving through Promotion of Life Cycle Assessment in Buildings*. ENSLIC BUILDING.
- Esnouf, A., Heijungs, R., Coste, G., Latrille, É., Steyer, J.-P., & Helias, A. (2019). A tool to guide the selection of impact categories for LCA studies by using the representativeness index. *Science of The Total Environment*, 658, 768-776. doi:<https://doi.org/10.1016/j.scitotenv.2018.12.194>
- European Commission. (2003). *Communication from the Commission to the Council and the European Parliament - Integrated Product Policy - Building on Environmental Life-Cycle Thinking*. Brussels.
- European Committee for Standardization. (2011). *EN 15978:2011 - Sustainability of Construction Works - Assessment of Environmental Performance of Buildings - Calculation Method*. European Committee for Standardization. Retrieved from

- https://standards.cen.eu/dyn/www/f?p=204:110:0:::FSP_PROJECT:31325&cs=16BA443169318FC086C4652D797E50C47
- Fauzi, R., Lavoie, P., Sorelli, L., Heidari, M., & Amor, B. (2019). Exploring the Current Challenges and Opportunities of Life Cycle Sustainability Assessment. *Sustainability*(11 (3)), 1-17. doi:<https://doi.org/10.3390/su11030636>
- Filimonau, V. (2016). Chapter 2: The Life Cycle Thinking Approach and the Method of Life Cycle Assessment (LCA). In *Life Cycle Assessment (LCA) and Life Cycle Analysis in Tourism: A Critical Review of Applications and Implications* (pp. 9-43). Springer International Publishing.
- Finnveden, G., Hauschild, M., Ekvall, T., Guinée, J., Heijungs, R., Hellweg, S., . . . Sangwon, S. (2009). Recent developments in Life Cycle Assessment. *Journal of Environmental Management* (91), 1-21. doi:<https://doi.org/10.1016/j.jenvman.2009.06.018>
- Fouche, M., & Crawford, R. (2017). Towards an integrated approach for evaluating both the life cycle environmental and financial performance of a building: A review. Sydney: Elsevier.
- Ghaffarian Hoseini, A., A. Zhanga, T., Nwadigo, O., Ghaffarian Hoseini, A., Naismith, N., Tookey, J., & Raahemifar, K. (2017). Application of nD BIM Integrated Knowledge-based Building Management System (BIM-IKBMS) for inspecting post-construction energy efficiency. *Renewable and Sustainable Energy Reviews*(72), 935-949. doi:<https://doi.org/10.1016/j.rser.2016.12.061>
- Gkatzios, M. (2018). *Integrating Building Information Modeling and Building Performance Simulation - A BIM-based simulation tool generating automated building energy performance analysis with explanatory feedback at the early design stages*. Eindhoven: University of Technology Eindhoven.
- Grilo, A., & Jardim-Goncalves, R. (2010). Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction*(19), 522-530. doi:<https://doi.org/10.1016/j.autcon.2009.11.003>
- Guinée, J. (2002). Handbook on life cycle assessment operational guide to the ISO standards. *The International Journal of Life Cycle Assessment*, 158-166. doi:<https://doi.org/10.1007/BF02978897>
- Guinée, J. (2010). Life cycle assessment: past, present and future. *International Symposium on Life Cycle Assessment and Construction*.
- Guinée, J. (2016). Life Cycle Sustainability Assessment: What Is It and What Are Its Challenges? In R. Clift, & A. Druckman, *Taking Stock of Industrial Ecology* (pp. 45-68). Guildford: Springer. doi:<https://doi.org/10.1007/978-3-319-20571-7>
- Guinée, J., Heijungs, R., Huppes, G., Zamagni, A., Masoni, P., Buonamici, R., . . . Rydberg, T. (2011). Life Cycle Assessment: Past, Present, and Future. *Environmental Science & Technology*(45), 90-96. doi:<https://doi.org/10.1021/es101316v>
- Häkkinen, T., & Kiviniemi, A. (2008). Sustainable Building and BIM. *Proceedings of SB08 Conference*, (pp. 21-25). Melbourne.
- Heijungs, R. (2015). Topological network theory and its application to LCA and related industrial ecology tools. *Journal of Environmental Accounting and Management*(3(2)), 151-167. doi:<https://doi.org/10.5890/JEAM.2015.06.005>
- Heijungs, R., Goedkoop, M., Struijs, J., Effting, S., Sevenster, M., & Huppes, G. (2003). *Towards a life cycle impact assessment method which comprises category indicators at the midpoint and the endpoint level*.

- Heijungs, R., Settanni, E., & Guinée, J. (2013). Toward a computational structure for life cycle sustainability analysis: unifying LCA and LCC. *The International Journal of Life Cycle Assessment*(18(9)), 1722-1733. doi:<https://doi.org/10.1007/s11367-012-0461-4>
- Hu, M., Kleijn, R., Bozhilova-Kisheva, K., & Di Maio, F. (2013). An approach to LCSA: The case of concrete recycling. *The International Journal of Life Cycle Assessment*(18(9)), 1793-1803. doi:<http://doi.org/10.1007/s11367-013-0599-8>
- Hunkeler, D., & Rebitzer, G. (2005). The Future of Life Cycle Assessment. *The International Journal of Life Cycle Assessment*(10(5)), 305-308. doi:<http://dx.doi.org/10.1065/lca2005.09.001>
- Hunkeler, D., Lichtenwort, K., & Rebitzer, G. (2008). *Environmental Life Cycle Costing*. SETAC.
- Hunt, R., & Franklin, W. (1996). LCA - How it came about. *The International Journal of Life Cycle Assessment*(1), 4-7. doi:<https://doi.org/10.1007/BF02978624>
- Iacovidou, E., Purnell, P., & Lim, M. (2018). The use of smart technologies in enabling construction components reuse: a viable method or a problem creating solution? *Journal of Environmental Management*(216), 214-223. doi:<http://dx.doi.org/10.1016/j.jenvman.2017.04.093>
- International Standard Organization. (1997). *ISO 14040. Environmental management – life cycle assessment – principles and framework*.
- Ismail, A., Nahar, A., & Scherer, R. (2017). Application of graph databases and graph theory concepts for advanced analysing of BIM models based on IFC standard. *24th International Workshop on Intelligent Computing in Engineering (EG-ICE 2017)*, (pp. 1–12). Nottingham. Retrieved from <https://www.researchgate.net/publication/318600860>
- ISO. (2006). *ISO 14044. Environmental management — Life cycle assessment — Requirements and guidelines*. International Standard Organization. Retrieved from <https://www.iso.org/standard/38498.html>
- ISO. (2017). *ISO 15686-5:2017 - Buildings and Constructed Assets - Service-Life Planning - Part 5: Life-cycle Costing*. International Organization for Standardization. doi:<https://www.iso.org/standard/61148.html>
- Jalaei, F., & Jade, A. (2015). Integrating building information modeling (BIM) and LEED system at the conceptual design stage of sustainable buildings. *Sustainable Cities and Society*(18), 95-107. doi:<https://doi.org/10.1016/j.scs.2015.06.007>
- Jørgensen, A., Finkbeiner, M., Jørgensen, M., & Hauschild, M. (2010). Defining the baseline in social life cycle assessment. *The International Journal of Life Cycle Assessment*(15(4)), 376-384. doi:<http://doi.org/10.1007/s11367-010-0176-3>
- Jade, A., & Jalaei, F. (2013). Integrating building information modelling with sustainability to design building projects at the conceptual stage. *Building Simulation* (4), 429-444.
- Khasreen, M., Banfill, P., & Menzies, G. (2009). Life-Cycle Assessment and the Environmental Impact of Buildings: A Review. *Sustainability* 2009(1), 674-701. doi:<http://doi.org/10.3390/su1030674>
- Kim, H., & Holme, P. (2015). Network Theory Integrated Life Cycle Assessment for an Electric Power System. *Sustainability*(7(8)), 10961-10975. doi:<http://doi.org/10.3390/su70810961>
- Klöpper, W. (2006). The Role of SETAC in the Development of LCA. *The International Journal of Life Cycle Assessment*(11), 116-122. doi:<http://doi.org/10.1065/lca2006.04.019>
- Klöpper, W. (2008). Life cycle sustainability assessment of products. *The International Journal of Life Cycle Assessment*(13(2)), 89-95. doi:<http://doi.org/10.1065/lca2008.02.376>

- Koppula, S. (2013, May 4). *Implementing BIM during the Construction Phase of Highway Project*. Retrieved from <http://au.autodesk.com/au-online/classes-on-demand/class-catalog/2013/infrastructure-design-suite/cr2307#chapter=0>.
- Kossiakoff, A., Sweet, W., Seymour, S., & Biemer, S. (2011). *Systems Engineering Principles and Practice*. New Jersey: John Wiley & Sons, Inc.
- Krigsvoll, G. (2007). Life Cycle Costing as part of decision making - use of building information models. In L. Bragança, M. Pinheiro, S. Jalali, R. Mateus, R. Amoêda, & M. Guedes, *Sustainable Construction, Materials and Practices* (pp. 433 - 440). Rotterdam: SINTEF Building and Infrastructure. Retrieved from <http://www.irbnet.de/daten/iconda/CIB11677.pdf>
- Langdon, D. (2007). *Life cycle costing (LCC) as a contribution to sustainable construction - Guidance on the use of the LCC Methodology and its application in public procurement*. Davis Langdon Management Consulting.
- Life Cycle Initiative. (2013). *The Methodological Sheets for Sub-categories in Social Life Cycle Assessment (S-LCA)*. United Nations Environment Programme and SETAC.
- Liu, S., Meng, Z., & Tam, C. (2015). Building information modeling based building design optimization for sustainability. *Energy Build* (105), 139-153.
- Llatas, C., Angulo Fornos, R., Bizcocho, N., Cortés Albalá, I., Falcón Ganfornina, R., Galeana, I., . . . Soust-Verdaguer, B. (2019). Towards a Life Cycle Sustainability Assessment method for the quantification and reduction of impacts of buildings life cycle. *IOP Conference Series: Earth and Environmental Science*(323(1)). doi:<https://doi.org/10.1088/1755-1315/323/1/012107>
- Macombe, C., Leskinen, P., Feschet, P., & Antikainen, R. (2013). Social life cycle assessment of biodiesel production at three levels: a literature review and development needs. *Journal of Cleaner Production*(Volume 52), 205-216. doi:<https://doi.org/10.1016/j.jclepro.2013.03.026>
- Markard, J., Raven, R., & Truffer, B. (2012). Sustainability Transitions: An Emerging Field of Research and its Prospects. *Research Policy* (41), 955-967.
- Matthews, H., Hendrickson, C., & Matthews, D. (2014). *Life Cycle Assessment: Quantitative Approaches for Decisions that Matter*. Retrieved from <https://csed.engin.umich.edu/assets/LCA-Book-Front-Matter-and-Chapter-1-01-07-16.pdf>
- McManus, M., & Taylor, C. (2015). The changing nature of life cycle assessment. *Biomass and Bioenergy* (82), 13-26. doi:<https://doi.org/10.1016/j.biombioe.2015.04.024>
- Meex, E., Hollberg, A., Knapen, E., Hildebrand, L., & Verbeeck, G. (2018). Requirements for applying LCA-based environmental impact assessment tools in the early stages of building design. *Building and Environment*, 228-236. doi:<https://doi.org/10.1016/j.buildenv.2018.02.016>
- Milieudatabase, N. (2019). *Information about the new NMD 3.0*. Retrieved from <https://milieudatabase.nl/nieuwe-bepalingsmethode-per-1-juli-2019/>
- Mithraratne, N., & Vale, B. (2004). Life cycle analysis model for New Zealand houses. *Building and Environment*, 483-492. doi:<https://doi.org/10.1016/j.buildenv.2003.09.008>
- Negele, H., Fricke, E., Schrepfer, L., & Härtlein, N. (1999). Modeling of Integrated Product Development Processes. *Proceedings of the 9th Annual Symposium of INCOSE* (pp. 1-9). Munich: Technical University of Munich.
- Norris, G. (2001(a)). Integrating Life Cycle Cost Analysis and LCA. *The International Journal of Life Cycle Assessment*(6(2)), 118-120. doi:<http://doi.org/10.1007/BF02977849>

- Norris, G. (2001(b)). Integrating Economic Analysis into LCA. *Environmental Quality Management*(10(3)), 59-64. doi:<https://doi.org/10.1002/tqem.1006>
- Nwodo, M., Anumba, C., & Asadi, S. (2017). BIM-Based Life Cycle Assessment and Costing of Buildings: Current Trends and Opportunities. *ASCE International Workshop on Computing in Civil Engineering*. doi:<https://doi.org/10.1061/9780784480847.007>
- Paavola, S., & Miettinen, R. (2018). Dynamics of design collaboration: BIM models as intermediary digital objects. *Computer Supported Cooperative Work*(28(1-2)), 1-26. doi:<http://doi.org/10.1007/s10606-018-9306-4>
- Parker, S. (2019). *Information on MoSCoW-method*. Retrieved from Parker Software: parkersoftware.com/blog/software-development-and-the-moscow-method/
- Petrillo, A., De Felice, F., Jannelli, E., Autorino, C., Minutillo, M., & Lavadera, A. (2016). Life cycle assessment (LCA) and life cycle cost (LCC) analysis model for a stand-alone hybrid renewable energy system. *Renewable Energy*(95), 337-355. doi:<https://doi.org/10.1016/j.renene.2016.04.027>
- Pinto, I. (2013). *Life Cycle Assessment: Overview and Application*. Toronto: Ryerson University. Retrieved from <https://digital.library.ryerson.ca/islandora/object/RULA%3A2401/datastream/OBJ/view>
- Rahmani Asl, M., Zarrinmehr, S., Bergin, M., & Yan, W. (2015). BPOpt: A framework for BIM-based performance optimization. *Energy and Buildings*(108), 401-412. doi:<https://doi.org/10.1016/j.enbuild.2015.09.011>
- Ramesh, T., Prakash, R., & Shukla, K. (2010). Life cycle energy analysis of buildings: An overview. *Energy and Buildings* (42), 1592-1600.
- Reddy, K. (2012). *BIM - Building Owners and Developers*. Hoboken: John Wiley & Sons, INC. Retrieved from https://uacg.bg/filebank/att_10831.pdf
- Rijksoverheid. (2016). *Nederland Circulair in 2050: Rijksbreed programma Circulaire Economie*. Rijksvastgoedbedrijf. (2019). *RVB BIM Specificatie V1.1-C*. Den Haag: Ministerie van Binnenlandse Zaken en Koninkrijksrelaties. Retrieved from <https://www.rijksvastgoedbedrijf.nl/expertise-en-diensten/b/building-information-modelling/documenten/richtlijn/2019/04/01/rvb-bim-specificatie-v1.1-c>
- Rijkswaterstaat & RIVM. (2015). *Beleidsverkenning circulaire economie in de bouw. Een perspectief voor de markt en overheid*. Rijkswaterstaat.
- Ristimäki, M., Säynäjoki, A., Heinonen, J., & Junnila, S. (2013). Combining life cycle costing and life cycle assessment for an analysis of a new residential district energy system design. *Energy*(63), 168-179. doi:<https://doi.org/10.1016/j.energy.2013.10.030>
- Röck, M., Hollberg, A., Habert, G., & Passer, A. (2018). LCA and BIM: Visualization of environmental potentials in building construction at early design stages. *Building Environment*. doi:<https://doi.org/10.1016/j.buildenv.2018.05.006>
- Röck, M., Passer, A., Ramon, D., & Allacker, K. (2019). The coupling of BIM and LCA - challenges identified through case study implementation. *Life-Cycle Analysis and Assessment in Civil Engineering: Towards an Integrated Vision*, 841-846.
- RVO. (2019, June). *Milieu Prestatie Gebouwen*. Retrieved from <https://www.rvo.nl/onderwerpen/duurzaam-ondernemen/gebouwen/wetten-en-regels-gebouwen/nieuwbouw/milieuprestatie-gebouwen>
- Sagar, A. (2019). *Reduce Memory Usage and Make Your Python Code Faster Using Generators*. Retrieved from <https://towardsdatascience.com/reduce-memory-usage-and-make-your-python-code-faster-using-generators-bd79dbfeb4c>

- Santos, R., Costa, A., Silvestre, J., & Pyl, L. (2019, July). Integration of LCA and LCC analysis within a BIM-based environment. *Automation in Construction*(103), 127-149. doi:<https://doi.org/10.1016/j.autcon.2019.02.011>
- Savino, M., Manzini, R., Della Selva, V., & Accorsi, R. (2017). A new model for environmental and economic evaluation of renewable energy systems: The case of wind turbines. *Applied Energy*(189), 739-752. doi:<https://doi.org/10.1016/j.apenergy.2016.11.124>
- Schlueter, A., & Thesseling, F. (2009). Building information model based energy/exergy performance assessment in early design stages. *Automation in Construction* (18)(18(2)), 153-163. doi:<https://doi.org/10.1016/j.autcon.2008.07.003>
- Schmidt, M. (2018). *Developing a framework for integrating life cycle environmental and economic assessment of buildings*. Melbourne: The University of Melbourne.
- Schwartz, Y., Raslan, R., & Mumovic, D. (2016). Implementing multi objective genetic algorithm for life cycle carbon footprint and life cycle cost minimisation: A building refurbishment case study. *Energy*(97), 58-68. doi:<https://doi.org/10.1016/j.energy.2015.11.056>
- Seo, S., Tucker, S., & Newton, P. (2007). Automated Material Selection and Environmental Assessment in the Context of 3D Building Modelling. *Journal of Green Building*(2), 51-61. doi:<https://doi.org/10.3992/jgb.2.2.51>
- SER. (2016). *Werken aan een Circulaire Economie: Geen Tijd te Verliezen*. Den Haag: Sociaal Economische Raad (SER). Retrieved from <https://www.ser.nl/-/media/ser/downloads/adviezen/2016/circulaire-economie.pdf>
- Settanni, E. (2008). The need for a computational structure of LCC. *The International Journal of Life Cycle Assessment*(13(7)), 526-531. doi:<http://doi.org/10.1007/s11367-008-0036-6>
- Shi, J., Liu, Z.-C., Zhang, H.-C., & Jiang, Q. (2015). Life Cycle Assessment: State of the Art and Future Perspectives. *Recent Patents on Mechanical Engineering*(8-3), 211-221. doi:<https://doi.org/10.2174/2212797608666150729231737>
- Shin, Y.-S., & Cho, K. (2015). BIM Application to Select Appropriate Design Alternative with Consideration of LCA and LCCA. *Mathematical Problems in Engineering*, 1-14. doi:<https://doi.org/10.1155/2015/281640>
- Siebert, A., Bezama, A., O'Keeffe, S., & Thrän, D. (2016). Social life cycle assessment indices and indicators to monitor the social implications of wood-based products. *Journal of Cleaner Production*(172), 4074-4084. doi:<https://doi.org/10.1016/j.jclepro.2017.02.146>
- Smith, P. (2014). BIM implementation - global strategies. *Creative Construction Conference*(85), 482-492. doi:<https://doi.org/10.1016/j.proeng.2014.10.575>
- Solihin, W., Eastman, C., & Lee, Y. (2016). A framework for fully integrated building information models in a federated environment. *Advanced Engineering Informatics*, 168-189. doi:<https://doi.org/10.1016/j.aei.2016.02.007>
- Soust-Verdaguer, B., Llatas, C., & García-Martínez, A. (2016). Simplification in life cycle assessment of single-family houses: A review of recent developments. *Building and Environment* (103), 215-227.
- Soust-Verdaguer, B., Llatas, C., & García-Martínez, A. (2017). Critical review of BIM-based LCA method to buildings. *Energy and Buildings* (136), 110-120.
- Soust-Verdaguer, B., Llatas, C., García-Martínez, A., & Gómez de Cózar, J. (2018). BIM-based LCA method to analyze envelope alternatives of single-family houses: case study in

- Uruguay. *Journal of Architectural Engineering*(24(3)). doi:[https://doi.org/10.1061/\(ASCE\)AE.1943-5568.0000303](https://doi.org/10.1061/(ASCE)AE.1943-5568.0000303)
- Starosta-Patyk, M. (2015). New products design decision making support by SimaPro software on the base of defective products management. *International Conference on Communication, Management and Information Technology (ICCMIT 2015)*(65), 1066-1074. doi:<https://doi.org/10.1016/j.procs.2015.09.051>
- Swarr, T., Hunkeler, D., Klöpffer, W., Pesonen, H.-L., Citroth, A., Brent, A., & Pagan, R. (2011). Environmental life-cycle costing: a code of practice. *International Journal of Life Cycle Assessment*(16(5)), 389–391. doi:<https://doi.org/10.1007/s11367-011-0287-5>
- Theiler, M., Bargstädt, H.-J., & Smarsly, K. (2018). IFC Monitor - An IFC schema extension for modeling structural health monitoring systems. *Advanced Engineering Informatics*(37), 54-65. doi:<https://doi.org/10.1016/J.AEI.2018.04.011>
- UNEP/SETAC. (2011). *Global Guidance Principles for Life Cycle Assessment Databases: A Basis for Greener Processes and Products*. UN.
- United Nations. (1987). *World Commission on Environment and Development - Our common future*. Oxford University Press.
- van Gemert, S. (2019). *Master Thesis - MPG Envie, A BIM-based LCA application for embodied impact assessment during the early design stages*. Eindhoven: TU Eindhoven.
- Verberne, J. (2016). *Building Circularity Indicators: An approach for measuring circularity of a building*. 2016: Eindhoven University of Technology.
- Weidema, B. (2003). *Market Information in Life Cycle Assessment*. Danish Environmental Protection Agency. Retrieved from https://www.researchgate.net/publication/259645192_Market_Information_in_Life_Cycle_Assessment
- Weisheng, L. (2017). Computational Building Information Modelling for construction waste management: Moving from rhetoric to reality. *Renewable and Sustainable Energy Reviews* (68), 587-595.
- Wittstock, B. G., Lenz, K., Saunders, T., Anderson, J., Carter, C., Gyetvai, Z., . . . Sjostrom, C. (2012). *EeBGuide Guidance Document - Part B: Buildings*. European Commission. Retrieved from https://www.eebguide.eu/eeblog/wp-content/uploads/2012/10/EeBGuide-B-FINAL-PR_2012-10-29.pdf
- Yee, P., Matta, C., Kam, C., Hagan, S., & Valdimarsson, O. (2009). *The GSA BIM Story*. United States General Services Administration. Retrieved from <https://hetnationaalbimplatform.nl/images/00%20onderzoek/the-gsa-bim-story.pdf>
- You, F., Tao, L., Graziano, D., & Snyder, S. (2011). Optimal design of sustainable cellulosic biofuel supply chains: Multiobjective optimization coupled with life cycle assessment and input-output analysis. *Process Systems Engineering*(52), 55-76. doi:<https://doi.org/10.1002/aic.12637>
- Zhiliang, M., Zhenhua, W., Wu, S., & Zhe, L. (2011). Application and extension of the IFC standard in construction cost estimating for tendering in China. *Automation in Construction* (20), 196-204.
- Zuo, J., Pullen, S., Rameezdeen, R., Bennetts, H., Wang, Y., Mao, G., . . . Duan, H. (2017). Green building evaluation from a life-cycle perspective in Australia: a critical review. *Renewable and Sustainable Energy Reviews*(70), 358-368. doi:<https://doi.org/10.1016/j.rser.2016.11.251>

APPENDICES

APPENDIX I - MPG-METHOD

The utilised version of the MPG-method was version 2.0. The following set of calculations are adopted from the MPG-method.

SCALING

The NMD-database uses standard dimensions of materials or products. If product dimensions are differing between the actual dimensions and the stated dimensions in the EPDs from the NMD-database, dimensions must be scaled. In other words, scaling must happen if the dimensions set in the design are differing between the standard dimensions and the design dimensions.

Within the MPG methodology, three scaling options are present: (i) *no scaling*; (ii) *mass scaling*; and (iii) *proportional scaling*. The first form, *no scaling*, means that no scaling can occur. This form means that the scaling factor (*Sfp*) is equal to the value 1. With regards to the *mass scaling*, the masses of individual product components are obtained from a specific product data supplier. However, since for this research the utilised NMD-database only uses generic *Category 3* data, the *Sfp* is set to 1. This set *Sfp* can be explained by that no specific product data suppliers are available in the *Category 3* data. Lastly, *proportional scaling* entails that the mass per functional unit is scaled to the same extent as the scalable dimensions.

Furthermore, a notion must be made with regards to the *proportional scaling*, because an extra scaling factor (*Sfo*) is required for each product component to indicate to what extent the various product components must scale. Since the generic data was used, no *Sfo* values were extracted from the NMD. If other data than the *Category 3* data is used, then the *Sfo* can be calculated. For the calculation of *Sfo*, the *Guidelines* of the MPG-methodology state that this value should be anywhere between 0 and 1. Since the *Sfo* could not be extracted related to the provided environmental data, the *Sfo* was set on 1, which means that all product components scale equally. Therefore, it must be stated a discrepancy can occur in results based on wrong scale factors. Altogether, the *Sftot* can be calculated as follows (Equation 5):

$$Sftot = Sfp \times Sfo \quad \text{Equation 5}$$

In Equation 5:

- Sfp* Scaling factor of a product, see Equation 6 or Equation 7;
- Sfo* Scaling factor of a product component, set to 1.

The *Sfp* of products that are scalable one dimension is calculated according to Equation 5 and Equation 8. If the products are scalable on two dimensions, *Sfp* is calculated following Equation 7, Equation 8 and Equation 9.

$$Sfp = \frac{Dim1; bw - vast1}{Dim1; def - vast1} \quad \text{Equation 6}$$

In Equation 6:

Dim1;bw	Dimension1 of a product in a building in millimetre;
Dim1;def	Default dimension1 of a product in millimetre;
vast1	Non-scalable component of dimension1 of a product in millimetre, see Equation 8.

$$Sfp = \frac{Dim1;bw - vast1}{Dim1;def - vast1} \times \frac{Dim2;bw - vast2}{Dim2;def - vast2} \quad \text{Equation 7}$$

In Equation 7:

Dim1;bw	Dimension1 of a product in a building in millimetre;
Dim1;def	Default dimension1 of a product in millimetre;
vast1	Non-scalable component of dimension1 of a product in millimetre, see Equation 8;
Dim2;bw	Dimension2 of a product in a building in millimetre;
Dim2;def	Default dimension2 of a product in millimetre;
vast2	Non-scalable component of dimension1 of a product in millimetre, see Equation 9.

$$vast1 = Dim1;def - Dim1sch \quad \text{Equation 8}$$

In Equation 8:

Dim1;def	Default dimension1 of a product in millimetre;
Dim1sch	Scalable component of dimension1 of a product in millimetre.

$$vast2 = Dim2;def - Dim2sch \quad \text{Equation 9}$$

In Equation 9:

Dim2;def	Default dimension2 of a product in millimetre;
Dim2sch	Scalable component of dimension2 of a product in millimetre.

THE ENVIRONMENTAL IMPACT OF A PRODUCT COMPONENT

The environmental impact of a product component is calculated by combining the environmental impact from production, transportation to the construction site and the combined profile for waste processing of a product. For the calculation of the environmental impact of a product component, some assumptions had to be made.

The assumptions that had to be made for the values of *Ogen* and *fTvol;o*. According to *Guideline 1* of the MPG-methodology, *Ogen* is a factor that must be added in the case that data is considered to be generic, such as the utilised data from *Category 3*. Depending on whether a specific document of material is consulted, the value of *Ogen* varies between 0.1 and 0.3. Since within the utilised data from the NMD-database do not contain documents mentioning *Ogen*, the value of *Ogen* was set on 0.3; the most unfavourable option.

Considering the $fTvol;o$ correction factor, this correction factor is based on the transport from a product component. Besides, this correction factor can be divided into two categories: (i) volume transport or (ii) mass transport. Mass transport is considered to be the least environmental friendly option, since within mass transport the maximum load and sizes of a lorry are taking into account, meaning that although more of a specific material can fit on a lorry considering its sizes, it does not mean that more material can be placed on the lorry based on the maximum load of that particular lorry, or vice versa. In other words, a lorry has a maximum permissible load available which cannot be exceeded by placing more material in terms of the available space on a lorry. As effect mass transport, more transport movements must occur, meaning that additional environmental impacts must be taken into account. On the other hand, volume transport means that certain materials only need a single transport movement according to their sizes and weight; thus, more environmentally friendly. Following *Guideline 2* of the MPG-methodology, keeping the least favourable mass transport into account, $fTvol;o$ should have the value of 1 when transported through mass transport. Additionally, keeping volume transport into account, the guideline also states that the value of $fTvol;o$ should lie between 0 and 1. Since the NMD-database does not specify which transportation option is used for the product components, $fTvol;o$ was set to a value of 1.

$$MEFeh = (1 + Ogen) \times \left(BMEFo + \frac{BMEFt;tc}{1000} \times tb \times fTvol;o + BMEFa;ac \right) \quad \text{Equation 10}$$

In Equation 10:

Ogen	Factor for unvalidated profiles, set to 0.3;
BMEFo	Environmental impact of the production of a product component;
BMEFt;tc	Environmental impact of transport of a product component;
tb	Transport distance to the construction site in kilometres, standard 150 km;
tc	Transport code for a type of transport, standard tc900;
fTvol;o	Correction factor for volume or mass transport of a product component, set to 1;
Ts/v/r/h/e	Transport distance for type of disposal scenario considering the waste for landfill (s), incineration (v), recycling (r), re-use (h) or own profile (e) in kilometres.

The environmental profile for waste processing ($BMEFa;ac$) for each product component can be calculated based on the applicable waste code (ac):

$$\begin{aligned} BMEFa;ac = & fs \times \left(BMEFac;s + \frac{BMEFt;tc}{1000} \times fTvol;o \times ts \right) + \dots \quad \text{Equation 11} \\ & \dots + fv \times \left(BMEFac;v + \frac{BMEFt;tc}{1000} \times fTvol;o \times tv \right) + \dots \\ & \dots + fr \times \left(BMEFac;r + \frac{BMEFt;tc}{1000} \times fTvol;o \times tr \right) + \dots \\ & \dots + fh \times \left(BMEFac;h + \frac{BMEFt;tc}{1000} \times fTvol;o \times th \right) + \dots \\ & \dots + fe \times \left(BMEFac;e + \frac{BMEFt;tc}{1000} \times fTvol;o \times te \right) \end{aligned}$$

In Equation 11:

fs/fv/fr/fh/fe	Waste of landfill (<i>s</i>), incineration (<i>v</i>), recycling (<i>r</i>), re-use (<i>h</i>) or own profile (<i>e</i>) in percentages;
BMEFac;i	Environmental impact of type of disposal (<i>i</i>) waste code (<i>ac</i>); waste for landfill (<i>s</i>), incineration (<i>v</i>), recycling (<i>r</i>), re-use (<i>h</i>) or own profile (<i>e</i>) in kg eq. per kg of material;
BMEFt;tc	Environmental impact of transport scenario <i>tc</i> in kg eq. per tkm of material;
tc	Transport code for the type of transport;
fTvol;o	Correction factor for volume or mass transport of a product component, set to 1;
ts/tv/tr/th/te	Transport distance for type of disposal scenario considering the waste for landfill (<i>s</i>), incineration (<i>v</i>), recycling (<i>r</i>), re-use (<i>h</i>) or own profile (<i>e</i>) in kilometres.

THE ENVIRONMENTAL IMPACT OF A PRODUCT

For the calculation of the environmental impact of a product, the factor with regards to the lifespan of a product in a building (*LDf*) must be determined. This lifespan can be calculated as follows:

$$LDreken = \text{minimum} (LDp; LDg)$$

Equation 12

In Equation 12:

LDp	Life span of a product in years;
LDg	Life span of a building in years, considering residential (75 years) or utilitarian buildings (50 years).

$$LDf = \text{maximum} \left(1; \frac{LDreken}{LDo} \right) - 1$$

Equation 13

In Equation 13:

LDreken	Life span of a product in a building in years;
LDo	Life span of a products component in years.

After the *LDf* has been calculated, the calculation of the environmental impact of a product can be made. A product in a building follows the different stages of a life-cycle, mentioned in Figure 3. By combining the environmental impact of product components (*MEfo*) following the different stages of a products lifecycle, an overall environmental impact of a product (*MEfp*) can be calculated. The following calculations form the determination of the environmental impact of a product.

$$MEfo = hvh \times Sftot \times (1 + Bafv) \times LDf \times MEfeh$$

Equation 14

In Equation 14:

hvh	Quantity of the product components;
Sftot	Scaling factor for the product component, see Equation 5;
Bafv	Construction waste in percentages;
LDf	Factor for a product component, see Equation 13;
MEFeh	Environmental impact of a product component, see Equation 10.

$$MEfp = \sum MEfo$$

Equation 15

In Equation 15:

MEfo	Environmental impact of product components, see Equation 14;
------	--

$$MEfp; i = \text{maximum}\left(1; \frac{LDg}{LDp}\right) * MEfp$$

Equation 16

In Equation 16:

LDp	Life span of a product in years;
LDg	Life span of a building in years, considering residential (75 years) or utilitarian buildings (50 years);
MEfp	Environmental impact of a product.

THE SHADOW COST OF A PRODUCT

The shadow cost of a product is calculated for every environmental impact category. The shadow cost of one impact category must be calculated based on fixed values, stated in Table 2. The following equation allows for the total shadow costs of a product for one impact category based on every product component in a building:

$$SKp = SKe \times MEfp; i$$

Equation 17

In Equation 17:

SKe	Shadow costs per equivalence unit, see Table 2;
MEfp;i	Environmental impact of a product over the entire lifespan of the building.

APPENDIX II - LIST OF INTERVIEWEES

Name	Company	Function	Date	Location	Subject(s)
Internal company					
Austin Krekels	Van de Ven	CEO	23-09-2019	Tilburg	Sustainability, Costing, Early design stages, BIM
Koen van der Vorst	Van de Ven	Project engineer	25-09-2019	Tilburg	Sustainability, Costing, VICO
Harry Nieuwkoop	Van Agtmaal	Project manager	28-09-2019	Oudenbosch	Sustainability, Costing, Early design stages, BIM
Marc Jansen	Van de Ven	Project manager	25-09-2019	Tilburg	Sustainability, Costing, Early design stages
Koen Jansens	Van de Ven	Project engineer	25-09-2019	Tilburg	Costing, VICO
Mark van Zelst	Van de Ven	Cost estimator	25-09-2019	Tilburg	Costing, VICO
Sander van Remortel	Van de Ven	Innovation Manager	18-09-2019 4-11-2019	Tilburg	BIM
External company					
Sander van Gemert	Arcadis	Cost and Project Manager	01-10-2019 30-10-2019	Arnhem	Sustainability, MPG-ENVIE, Costing, BIM
Ivando Gondim Lossio Junior	TU Eindhoven	PdEng-student	16-10-2019	Eindhoven	Sustainability, Circularity, BIM
Nuo Zhang	TU Eindhoven	PhD-student	16-10-2019	Eindhoven	Sustainability, Circularity, BIM

APPENDIX III - PYTHON SCRIPTS

MAINAPP.PY

```
# This Python file uses the following encoding: utf-8

# Imports of the building
import sys
import os
import matplotlib.pyplot as plt
from matplotlib.backends.backend_qt4agg import FigureCanvasQTAgg as FigureCanvas
import seaborn as sns
import ifcopenshell
import ifcopenshell.geom
settings = ifcopenshell.geom.settings()
settings.set(settings.USE_PYTHON_OPENCASCADE, True)
from OCC.Display.backend import load_backend
load_backend("qt-pyqt4")
import openpyxl
import datetime
import os.path
import pandas as pd
import numpy as np
from PyQt4 import QtGui, QtCore
from PyQt4.QtGui import QLineEdit
from gui import Ui_MainWindow
import importlib
import importcomponents

# Configures functionality of life span of building
class LifeSpanBuilding:
    def __init__(self, building_type):
        self.building_type = building_type

    def LDg(self):
        if self.building_type == "Residential":
            return 75
        elif self.building_type == "Utilitarian":
            return 50

### CONFIGURES FUNCTIONALITY OF THE APPLICATION ###
# Configures functionality of GUI.
class Main(QtGui.QMainWindow, LifeSpanBuilding):

    def __init__(self):
        QtGui.QMainWindow.__init__(self)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

# Configures 'Open-Save-Close' functionality.
self.file_path = None
self.ui.actionOpen.triggered.connect(self.open_IFC)
self.ui.actionSave.setEnabled(False)
self.ui.actionSave.triggered.connect(self.save_plots)
self.ui.actionClose.triggered.connect(self.close_application)
self.ui.btn_OpenIFC.clicked.connect(self.open_IFC)
self.ui.btn_Calculate.clicked.connect(self.overview_tab)
self.ui.btn_Generate.setEnabled(False)
self.ui.btn_Generate.clicked.connect(self.overview_tab)
self.ui.slider.setEnabled(False)
self.ui.slider.valueChanged.connect(self.slider_change)
self.slideredit = QLineEdit()

# Configure 'GFA Input' functionality
regex = QtCore.QRegExp("[0-9_]+")
validator = QtGui.QRegExpValidator(regex)
self.ui.GFAInput.setValidator(validator)

# Configure 'Overview' functionality
self.y_values1 = None
self.figure = plt.figure()
self.canvas1 = FigureCanvas(self.figure)
```

```

layout = QtGui.QVBoxLayout()
layout.addWidget(self.canvas1)
self.ui.overviewTab.setLayout(layout)

def overview_tab(self):
    # Enables the 'Save' button when the 'Calculate' button is clicked.
    self.ui.actionSave.setEnabled(True)
    self.ui.btn_Generate.setEnabled(True)
    self.ui.slider.setEnabled(True)

    textGenerate = self.slideredit.text()
    if textGenerate == "":
        textGenerate = 5
    valueGenerate = int(textGenerate)
    factorecoimpact = (valueGenerate - 1) / 8
    factorenvimpact = 1 - factorecoimpact

# Variable that stores the chosen building type
    building_type = self.ui.buildingTypeChoice.currentText()

# Variable that stores the chosen calculation type
    calculation_choice = self.ui.calculationTypeChoice.currentText()
    self.calculation_choice = calculation_choice

# Variable that stores the chosen building system
    building_system = self.ui.buildingSystem.currentText()
    self.building_system = building_system

# Gross floor area (GFA) of the building
    GFA = self.ui.GFAInput.text()

# Configures functionality of basevalues.py
    if building_type == "Choose...":
        LDg = 0
    elif calculation_choice == "Choose...":
        LDg = 0
    elif building_type == "Residential":
        LDg = 75
    elif building_type == "Utilitarian":
        LDg = 50
    filename = 'basevalues.py'
    with open(filename, 'w') as f:
        f.write('### BASE VALUES ###' + "\n")
        f.write('building_type = {}'.format(building_type) + "'" + "\n")
        f.write('calculation_choice = {}'.format(calculation_choice) + "'" + "\n")
        f.write('building_system = {}'.format(building_system) + "'" + "\n")
        f.write('factorenvimpact = {}'.format(factorenvimpact) + " + "\n")
        f.write('factorecoimpact = {}'.format(factorecoimpact) + " + "\n")
        f.write('LDg = {}'.format(LDg) + "\n")
        f.write('tc = "900t"' + "\n")
        f.write('ffVol = 1' + "\n")
        f.write('Ts = 100' + "\n")
        f.write('Tv = 100' + "\n")
        f.write('Tr = 50' + "\n")
        f.write('Th = 0' + "\n")
        f.write('Te = 100' + "\n")
        f.write('Ogen = 0.3' + "\n")
    f.close()
    importlib.reload(importcomponents)
    CostInteriorWallGeneric = 0
    CostInteriorWallSpecific = 0
    InteriorWallSurfaceGeneric = 0
    InteriorWallSurfaceSpecific = 0
    CostFloorGeneric = 0
    CostFloorSpecific = 0
    FloorSurfaceGeneric = 0
    FloorSurfaceSpecific = 0
    CostExteriorWallGeneric = 0
    CostExteriorWallSpecific = 0
    ExteriorWallSurfaceGeneric = 0
    ExteriorWallSurfaceSpecific = 0

```

```

TotalCostGeneric = 0
TotalCostSpecific = 0
dictionary_SCp = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
dictionary_MEFp = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Environmental_InteriorWall = "-"
Environmental_Floor = "-"
Environmental_ExteriorWall = "-"
Economic_InteriorWall = "-"
Economic_Floor = "-"
Economic_ExteriorWall = "-"
Overall_InteriorWall = "-"
Overall_Floor = "-"
Overall_ExteriorWall = "-"
TotalSCp = 0
SCp_BD2012 = 0
if building_type != "Choose...":
    if calculation_choice != "Choose...":
        importlib.reload(importcomponents)
        CostInteriorWallGeneric = importcomponents.CostWallGeneric_InteriorWall
        CostInteriorWallSpecific = importcomponents.CostWallSpecific_InteriorWall
        InteriorWallSurfaceGeneric = importcomponents.WallSurfaceGeneric_InteriorWall
        InteriorWallSurfaceSpecific = importcomponents.WallSurfaceSpecific_InteriorWall
        CostFloorGeneric = importcomponents.CostFloorGeneric_Floor
        CostFloorSpecific = importcomponents.CostFloorSpecific_Floor
        FloorSurfaceGeneric = importcomponents.FloorSurfaceGeneric_Floor
        FloorSurfaceSpecific = importcomponents.FloorSurfaceSpecific_Floor
        CostExteriorWallGeneric = importcomponents.CostWallGeneric_ExteriorWall
        CostExteriorWallSpecific = importcomponents.CostWallSpecific_ExteriorWall
        ExteriorWallSurfaceGeneric = importcomponents.WallSurfaceGeneric_ExteriorWall
        ExteriorWallSurfaceSpecific = importcomponents.WallSurfaceSpecific_ExteriorWall
        TotalCostGeneric = importcomponents.TotalCostGeneric
        TotalCostSpecific = importcomponents.TotalCostSpecific
        dictionary_SCp = importcomponents.dictionary_SCp
        dictionary_MEFp = importcomponents.dictionary_MEFp
        TotalSCp = importcomponents.TotalSCp
        Environmental_InteriorWall = importcomponents.bestchoice_envimp_InteriorWall_type
        Environmental_Floor = importcomponents.bestchoice_envimp_Floor_type
        Environmental_ExteriorWall = importcomponents.bestchoice_envimp_ExteriorWall_type
        Economic_InteriorWall = importcomponents.bestchoice_ecoimp_InteriorWall_type
        Economic_Floor = importcomponents.bestchoice_ecoimp_Floor_type
        Economic_ExteriorWall = importcomponents.bestchoice_ecoimp_ExteriorWall_type
        Overall_InteriorWall = importcomponents.bestchoice_overall_InteriorWall_type
        Overall_Floor = importcomponents.bestchoice_overall_Floor_type
        Overall_ExteriorWall = importcomponents.bestchoice_overall_ExteriorWall_type

    if GFA == "":
        GFA = FloorSurfaceGeneric + FloorSurfaceSpecific
    elif GFA == 0:
        GFA = 1.0E-10

    if TotalSCp > 0:
        SCp_BD2012 = (TotalSCp) / int(GFA) / int(LDg)
    else:
        SCp_BD2012 = 0

# Values related to the choicecalculation.
CostInteriorWallGeneric = u"\N{euro sign}" + str(round(CostInteriorWallGeneric, 2))
CostInteriorWallSpecific = u"\N{euro sign}" + str(round(CostInteriorWallSpecific, 2))
InteriorWallSurfaceGeneric = str(round(InteriorWallSurfaceGeneric)) + " m2"
InteriorWallSurfaceSpecific = str(round(InteriorWallSurfaceSpecific)) + " m2"
CostFloorGeneric = u"\N{euro sign}" + str(round(CostFloorGeneric, 2))
CostFloorSpecific = u"\N{euro sign}" + str(round(CostFloorSpecific, 2))
FloorSurfaceGeneric = str(round(FloorSurfaceGeneric)) + " m2"
FloorSurfaceSpecific = str(round(FloorSurfaceSpecific)) + " m2"
CostExteriorWallGeneric = u"\N{euro sign}" + str(round(CostExteriorWallGeneric, 2))
CostExteriorWallSpecific = u"\N{euro sign}" + str(round(CostExteriorWallSpecific, 2))
ExteriorWallSurfaceGeneric = str(round(ExteriorWallSurfaceGeneric)) + " m2"
ExteriorWallSurfaceSpecific = str(round(ExteriorWallSurfaceSpecific)) + " m2"
TotalCostGeneric = u"\N{euro sign}" + str(round(TotalCostGeneric, 2))
TotalCostSpecific = u"\N{euro sign}" + str(round(TotalCostSpecific, 2))
MEFp_GWP = str(round(dictionary_MEFp[0], 2))

```

```

MEFp_ODP = str(round(dictionary_MEFp[1], 2))
MEFp_POCP = str(round(dictionary_MEFp[2], 2))
MEFp_AP = str(round(dictionary_MEFp[3], 2))
MEFp_EP = str(round(dictionary_MEFp[4], 2))
MEFp_HTP = str(round(dictionary_MEFp[5], 2))
MEFp_FAETP = str(round(dictionary_MEFp[6], 2))
MEFp_MAETP = str(round(dictionary_MEFp[7], 2))
MEFp_TETP = str(round(dictionary_MEFp[8], 2))
MEFp_ADPf = str(round(dictionary_MEFp[9], 2))
MEFp_ADPc = str(round(dictionary_MEFp[10], 2))
SCp_GWP = str(round(dictionary_SCp[0], 2))
SCp_ODP = str(round(dictionary_SCp[1], 2))
SCp_POCP = str(round(dictionary_SCp[2], 2))
SCp_AP = str(round(dictionary_SCp[3], 2))
SCp_EP = str(round(dictionary_SCp[4], 2))
SCp_HTP = str(round(dictionary_SCp[5], 2))
SCp_FAETP = str(round(dictionary_SCp[6], 2))
SCp_MAETP = str(round(dictionary_SCp[7], 2))
SCp_TETP = str(round(dictionary_SCp[8], 2))
SCp_ADPf = str(round(dictionary_SCp[9], 2))
SCp_ADPc = str(round(dictionary_SCp[10], 2))
SCp_final = "€ " + str(round(TotalSCp, 2))

# Return values to the application
self.ui.labelEnvironmental_InteriorWall.setText(Environmental_InteriorWall)
self.ui.labelEnvironmental_Floor.setText(Environmental_Floor)
self.ui.labelEnvironmental_ExteriorWall.setText(Environmental_ExteriorWall)
self.ui.labelEconomic_InteriorWall.setText(Economic_InteriorWall)
self.ui.labelEconomic_Floor.setText(Economic_Floor)
self.ui.labelEconomic_ExteriorWall.setText(Economic_ExteriorWall)
self.ui.labelOverall_InteriorWall.setText(Overall_InteriorWall)
self.ui.labelOverall_Floor.setText(Overall_Floor)
self.ui.labelOverall_ExteriorWall.setText(Overall_ExteriorWall)
self.ui.labelCostInteriorWallGeneric.setText(CostInteriorWallGeneric)
self.ui.labelCostInteriorWallSpecific.setText(CostInteriorWallSpecific)
self.ui.labelInteriorWallSurfaceGeneric.setText(InteriorWallSurfaceGeneric)
self.ui.labelInteriorWallSurfaceSpecific.setText(InteriorWallSurfaceSpecific)
self.ui.labelCostFloorGeneric.setText(CostFloorGeneric)
self.ui.labelCostFloorSpecific.setText(CostFloorSpecific)
self.ui.labelFloorSurfaceGeneric.setText(FloorSurfaceGeneric)
self.ui.labelFloorSurfaceSpecific.setText(FloorSurfaceSpecific)
self.ui.labelCostExteriorWallGeneric.setText(CostExteriorWallGeneric)
self.ui.labelCostExteriorWallSpecific.setText(CostExteriorWallSpecific)
self.ui.labelExteriorWallSurfaceGeneric.setText(ExteriorWallSurfaceGeneric)
self.ui.labelExteriorWallSurfaceSpecific.setText(ExteriorWallSurfaceSpecific)
self.ui.labelTotalCost_empty.setText("-")
self.ui.labelTotalCost_empty_2.setText("-")
self.ui.labelTotalCostGeneric.setText(TotalCostGeneric)
self.ui.labelTotalCostSpecific.setText(TotalCostSpecific)
self.ui.labelADPc.setText(MEFp_ADPc)
self.ui.labelADPf.setText(MEFp_ADPf)
self.ui.labelGWP.setText(MEFp_GWP)
self.ui.labelODP.setText(MEFp_ODP)
self.ui.labelPOCP.setText(MEFp_POCP)
self.ui.labelAP.setText(MEFp_AP)
self.ui.labelEP.setText(MEFp_EP)
self.ui.labelHTP.setText(MEFp_HTP)
self.ui.labelFAETP.setText(MEFp_FAETP)
self.ui.labelMAETP.setText(MEFp_MAETP)
self.ui.labelTETP.setText(MEFp_TETP)
self.ui.labelADPc_SC.setText(SCp_ADPc)
self.ui.labelADPf_SC.setText(SCp_ADPf)
self.ui.labelGWP_SC.setText(SCp_GWP)
self.ui.labelODP_SC.setText(SCp_ODP)
self.ui.labelPOCP_SC.setText(SCp_POCP)
self.ui.labelAP_SC.setText(SCp_AP)
self.ui.labelEP_SC.setText(SCp_EP)
self.ui.labelHTP_SC.setText(SCp_HTP)
self.ui.labelFAETP_SC.setText(SCp_FAETP)
self.ui.labelMAETP_SC.setText(SCp_MAETP)
self.ui.labelTETP_SC.setText(SCp_TETP)

```

```

self.ui.labelTotalEI_SC.setText(SCp_final)

# Building Decree 2012 test.
if SCp_BD2012 == 0:
    self.ui.labelBC2012_ApprovedDeclined.setText("")
elif 0 <= SCp_BD2012 <= 1:
    self.ui.labelBC2012_ApprovedDeclined.setText("<font color='green'>Approved<font color='green'>")
    SCp_BD2012 = u"\N{euro sign}" + str(round(SCp_BD2012, 2))
    self.ui.labelBC2012_SCFinal.setText(SCp_BD2012)
elif SCp_BD2012 > 1:
    self.ui.labelBC2012_ApprovedDeclined.setText("<font color='red'>Declined<font color='red'>")
    SCp_BD2012 = u"\N{euro sign}" + str(round(SCp_BD2012, 2))
    self.ui.labelBC2012_SCFinal.setText(SCp_BD2012)

# Configures functionality of bar charts
plt.clf()
importlib.reload(importcomponents)
sns.set_style("darkgrid")
bar_width = 0.35
positionbar = ((1 - bar_width) - bar_width) / 2 + bar_width
opacity = 0.1
opacity_full = 0.8

interiorwall_labels = importcomponents.dictionary_Type_list_InteriorWall
floor_labels = importcomponents.dictionary_Type_list_Floor
exteriorwall_labels = importcomponents.dictionary_Type_list_ExteriorWall
interiorwall_SCp_results_generic = importcomponents.dictionary_SCp_g_InteriorWall_totals
interiorwall_SCp_results_specific = importcomponents.dictionary_SCp_s_InteriorWall_totals
interiorwall_C_results_generic = importcomponents.dictionary_CostWall_g_InteriorWall
interiorwall_C_results_specific = importcomponents.dictionary_CostWall_s_InteriorWall

floor_SCp_results_generic = importcomponents.dictionary_SCp_g_Floor_totals
floor_SCp_results_specific = importcomponents.dictionary_SCp_s_Floor_totals
floor_C_results_generic = importcomponents.dictionary_CostFloor_g_Floor
floor_C_results_specific = importcomponents.dictionary_CostFloor_s_Floor

exteriorwall_SCp_results_generic = importcomponents.dictionary_SCp_g_ExteriorWall_totals
exteriorwall_SCp_results_specific = importcomponents.dictionary_SCp_s_ExteriorWall_totals
exteriorwall_C_results_generic = importcomponents.dictionary_CostWall_g_ExteriorWall
exteriorwall_C_results_specific = importcomponents.dictionary_CostWall_s_ExteriorWall

interiorwall_index = np.arange(len(interiorwall_labels))
floor_index = np.arange(len(floor_labels))
exteriorwall_index = np.arange(len(exteriorwall_labels))

environmental_label = "Generic environmental impact"
environmentaladdition_label = "Total environmental impact"
environmentaloverall_label = "Specific environmental impact"
economic_label = "Generic economic impact"
economicaddition_label = "Total economic impact"
economicoverall_label = "Specific economic impact"
environmental_labelcolor = 'forestgreen'
economic_labelcolor = 'navy'

if calculation_choice == "Specific":
    interiorwall_SCp_results_specific_totals = sum(interiorwall_SCp_results_specific)
    interiorwall_SCp_results_specific_totals_show = [interiorwall_SCp_results_specific_totals for i in interiorwall_SCp_results_generic]
    interiorwall_SCp_results_specific_add = [interiorwall_SCp_results_specific_totals + i for i in interiorwall_SCp_results_generic]
    interiorwall_SCp_results_generic_totals_show = interiorwall_SCp_results_specific_add
    interiorwall_SCp_results_specific_add_percentage = [x / y for x, y in zip(interiorwall_SCp_results_generic,
interiorwall_SCp_results_specific_add)]
    interiorwall_SCp_results_specific_add_percentage = [100.0 * i for i in interiorwall_SCp_results_specific_add_percentage]
    interiorwall_SCp_results_specific_add_percentage = np.round(interiorwall_SCp_results_specific_add_percentage, 1)
    interiorwall_SCp_results_specific_add_percentage = interiorwall_SCp_results_specific_add_percentage.tolist()
    interiorwall_SCp_results_specific_add_percentage = ["+" + str(x) + "%" for x in interiorwall_SCp_results_specific_add_percentage]
    if len(interiorwall_labels) == 1:
        interiorwall_C_results_specific_totals = interiorwall_C_results_specific
    elif len(interiorwall_labels) > 1:
        interiorwall_C_results_specific_totals = sum(interiorwall_C_results_specific)
    interiorwall_C_results_specific_totals_show = [interiorwall_C_results_specific_totals for i in interiorwall_C_results_generic]
    interiorwall_C_results_specific_add = [interiorwall_C_results_specific_totals + i for i in interiorwall_C_results_generic]

```



```

interiorwall_C_results_generic_totals_show = interiorwall_C_results_specific_add
interiorwall_C_results_specific_add_percentage = [x / y for x, y in zip(interiorwall_C_results_generic,
interiorwall_C_results_specific_add)]
interiorwall_C_results_specific_add_percentage = [100.0 * i for i in interiorwall_C_results_specific_add_percentage]
interiorwall_C_results_specific_add_percentage = np.round(interiorwall_C_results_specific_add_percentage, 1)
interiorwall_C_results_specific_add_percentage = interiorwall_C_results_specific_add_percentage.tolist()
interiorwall_C_results_specific_add_percentage = ["+" + str(x) + "%" for x in interiorwall_C_results_specific_add_percentage]

floor_SCp_results_specific_totals = sum(floor_SCp_results_specific)
floor_SCp_results_specific_totals_show = [floor_SCp_results_specific_totals for i in floor_SCp_results_generic]
floor_SCp_results_specific_add = [floor_SCp_results_specific_totals + i for i in floor_SCp_results_generic]
floor_SCp_results_generic_totals_show = floor_SCp_results_specific_add
floor_SCp_results_specific_add_percentage = [x / y for x, y in zip(floor_SCp_results_generic, floor_SCp_results_specific_add)]
floor_SCp_results_specific_add_percentage = [100.0 * i for i in floor_SCp_results_specific_add_percentage]
floor_SCp_results_specific_add_percentage = np.round(floor_SCp_results_specific_add_percentage, 1)
floor_SCp_results_specific_add_percentage = floor_SCp_results_specific_add_percentage.tolist()
floor_SCp_results_specific_add_percentage = ["+" + str(x) + "%" for x in floor_SCp_results_specific_add_percentage]
if len(floor_labels) == 1:
    floor_C_results_specific_totals = floor_C_results_specific
elif len(floor_labels) > 1:
    floor_C_results_specific_totals = sum(floor_C_results_specific)
    floor_C_results_specific_totals_show = [floor_C_results_specific_totals for i in floor_C_results_generic]
    floor_C_results_specific_add = [floor_C_results_specific_totals + i for i in floor_C_results_generic]
    floor_C_results_generic_totals_show = floor_C_results_specific_add
    floor_C_results_specific_add_percentage = [x / y for x, y in zip(floor_C_results_generic, floor_C_results_specific_add)]
    floor_C_results_specific_add_percentage = [100.0 * i for i in floor_C_results_specific_add_percentage]
    floor_C_results_specific_add_percentage = np.round(floor_C_results_specific_add_percentage, 1)
    floor_C_results_specific_add_percentage = floor_C_results_specific_add_percentage.tolist()
    floor_C_results_specific_add_percentage = ["+" + str(x) + "%" for x in floor_C_results_specific_add_percentage]

exteriorwall_SCp_results_specific_totals = sum(exteriorwall_SCp_results_specific)
exteriorwall_SCp_results_specific_totals_show = [exteriorwall_SCp_results_specific_totals for i in exteriorwall_SCp_results_generic]
exteriorwall_SCp_results_specific_add = [exteriorwall_SCp_results_specific_totals + i for i in exteriorwall_SCp_results_generic]
exteriorwall_SCp_results_generic_totals_show = exteriorwall_SCp_results_specific_add
exteriorwall_SCp_results_specific_add_percentage = [x / y for x, y in zip(exteriorwall_SCp_results_generic,
exteriorwall_SCp_results_specific_add)]
exteriorwall_SCp_results_specific_add_percentage = [100.0 * i for i in exteriorwall_SCp_results_specific_add_percentage]
exteriorwall_SCp_results_specific_add_percentage = np.round(exteriorwall_SCp_results_specific_add_percentage, 1)
exteriorwall_SCp_results_specific_add_percentage = exteriorwall_SCp_results_specific_add_percentage.tolist()
exteriorwall_SCp_results_specific_add_percentage = ["+" + str(x) + "%" for x in exteriorwall_SCp_results_specific_add_percentage]
if len(exteriorwall_labels) == 1:
    exteriorwall_C_results_specific_totals = exteriorwall_C_results_specific
elif len(exteriorwall_labels) > 1:
    exteriorwall_C_results_specific_totals = sum(exteriorwall_C_results_specific)
    exteriorwall_C_results_specific_totals_show = [exteriorwall_C_results_specific_totals for i in exteriorwall_C_results_generic]
    exteriorwall_C_results_specific_add = [exteriorwall_C_results_specific_totals + i for i in exteriorwall_C_results_generic]
    exteriorwall_C_results_generic_totals_show = exteriorwall_C_results_specific_add
    exteriorwall_C_results_specific_add_percentage = [x / y for x, y in zip(exteriorwall_C_results_generic,
exteriorwall_C_results_specific_add)]
    exteriorwall_C_results_specific_add_percentage = [100.0 * i for i in exteriorwall_C_results_specific_add_percentage]
    exteriorwall_C_results_specific_add_percentage = np.round(exteriorwall_C_results_specific_add_percentage, 1)
    exteriorwall_C_results_specific_add_percentage = exteriorwall_C_results_specific_add_percentage.tolist()
    exteriorwall_C_results_specific_add_percentage = ["+" + str(x) + "%" for x in exteriorwall_C_results_specific_add_percentage]

else:
    interiorwall_SCp_results_generic_totals_show = interiorwall_SCp_results_generic
    interiorwall_C_results_generic_totals_show = interiorwall_C_results_generic
    floor_SCp_results_generic_totals_show = floor_SCp_results_generic
    floor_C_results_generic_totals_show = floor_C_results_generic
    exteriorwall_SCp_results_generic_totals_show = exteriorwall_SCp_results_generic
    exteriorwall_C_results_generic_totals_show = exteriorwall_C_results_generic

ax1 = self.figure.add_subplot(3, 2, 1)
def autolabel_intwall_env_generic(rects1):
    for idx, rect in enumerate(rects1):
        height = rect.get_height()
        ax1.text(rect.get_x() + rect.get_width() / 2, height, interiorwall_SCp_results_generic_totals_show[idx], fontsize='11', ha='left',
va='bottom')
def autolabel_intwall_env_specific(rects2):
    for idx, rect in enumerate(rects2):
        height = rect.get_height()

```

```

    ax1.text(rect.get_x() + rect.get_width() / 2, height, interiorwall_SCp_results_specific_add_percentage[idx], fontsize='10',
fontstyle='italic', ha='center', va='bottom')
    if calculation_choice == "Generic":
        rects1 = ax1.bar((interiorwall_index - (bar_width/2)), interiorwall_SCp_results_generic_totals_show, bar_width, alpha=opacity_full,
label=environmental_label, color=environmental_labelcolor)
        autolabel_intwall_env_generic(rects1)
        ylim_ax1 = max(interiorwall_SCp_results_generic) * 1.4
    if calculation_choice == "Specific":
        rects1 = ax1.bar((interiorwall_index - bar_width + 0.1), interiorwall_SCp_results_specific_totals_show, bar_width, alpha=opacity_full,
label=environmentaloverall_label, color=environmental_labelcolor)
        rects2 = ax1.bar((interiorwall_index - 0.1), interiorwall_SCp_results_specific_add, bar_width, alpha = opacity, label =
environmentaladdition_label, color = environmental_labelcolor)
        autolabel_intwall_env_specific(rects1)
        autolabel_intwall_env_generic(rects2)
        ylim_ax1 = max(interiorwall_SCp_results_specific_add) * 1.4

xlim_ax1_left = min(interiorwall_index) - positionbar
xlim_ax1_right = max(interiorwall_index) + 0.5
ax1.set_ylim(0, ylim_ax1)
ax1.set_xlim(xlim_ax1_left, xlim_ax1_right)
ax1.grid(b=None, axis='x')
ax1.tick_params(axis='both', labelsize='10', length=0)
ax1.set_title(label='Shadow costs per product - Interior Wall (%)' % u"\N{euro sign}", loc='center', fontsize='12', fontweight='bold')
ax1.set_xticks(interiorwall_index)
ax1.set_xticklabels(interiorwall_labels)
ax1.legend(prop={'size': 9}, framealpha=0.5)

ax2 = self.figure.add_subplot(3, 2, 2)
def autolabel_intwall_eco_generic(rects3):
    for idx, rect in enumerate(rects3):
        height = rect.get_height()
        ax2.text(rect.get_x() + rect.get_width() / 2, height, interiorwall_C_results_generic_totals_show[idx], fontsize='11', ha='left', va='bottom')
def autolabel_intwall_eco_specific(rects4):
    for idx, rect in enumerate(rects4):
        height = rect.get_height()
        ax2.text(rect.get_x() + rect.get_width() / 2, height,
        interiorwall_C_results_specific_add_percentage[idx], fontsize='10', fontstyle='italic', ha='center', va='bottom')
    if calculation_choice == "Generic":
        rects3 = ax2.bar((interiorwall_index - (bar_width / 2)), interiorwall_C_results_generic_totals_show, bar_width, alpha=opacity_full,
label=economic_label, color=economic_labelcolor)
        autolabel_intwall_eco_generic(rects3)
        ylim_ax2 = max(interiorwall_C_results_generic) * 1.4
    if calculation_choice == "Specific":
        rects3 = ax2.bar((interiorwall_index - bar_width + 0.1), interiorwall_C_results_specific_totals_show, bar_width, alpha=opacity_full,
label=economicoverall_label, color=economic_labelcolor)
        rects4 = ax2.bar((interiorwall_index - 0.1), interiorwall_C_results_specific_add, bar_width, alpha=opacity, label=economicaddition_label,
color=economic_labelcolor)
        autolabel_intwall_eco_specific(rects3)
        autolabel_intwall_eco_generic(rects4)
        ylim_ax2 = max(interiorwall_C_results_specific_add) * 1.4
    xlim_ax2_left = min(interiorwall_index) - positionbar
    xlim_ax2_right = max(interiorwall_index) + 0.5
    ax2.set_ylim(0, ylim_ax2)
    ax2.set_xlim(xlim_ax2_left, xlim_ax2_right)
    ax2.grid(b=None, axis='x')
    ax2.tick_params(axis='both', labelsize='10', length=0)
    ax2.set_title(label='Costs per product - Interior Wall (%)' % u"\N{euro sign}", loc='center', fontsize='12', fontweight='bold')
    ax2.set_xticks(interiorwall_index)
    ax2.set_xticklabels(interiorwall_labels)
    ax2.legend(prop={'size': 9}, framealpha=0.5)

ax3 = self.figure.add_subplot(3, 2, 3)
def autolabel_floor_env_generic(rects5):
    for idx, rect in enumerate(rects5):
        height = rect.get_height()
        ax3.text(rect.get_x() + rect.get_width() / 2, height, floor_SCp_results_generic_totals_show[idx], fontsize='11', ha='left', va='bottom')
def autolabel_floor_env_specific(rects6):
    for idx, rect in enumerate(rects6):
        height = rect.get_height()
        ax3.text(rect.get_x() + rect.get_width() / 2, height, floor_SCp_results_specific_add_percentage[idx], fontsize='10', fontstyle='italic',
ha='center', va='bottom')

```

```

if calculation_choice == "Generic":
    rects5 = ax3.bar((floor_index - (bar_width / 2)), floor_SCp_results_generic_totals_show, bar_width, alpha=opacity_full,
label=environmental_label, color=environmental_labelcolor)
    autolabel_floor_env_generic(rects5)
    ylim_ax3 = max(floor_SCp_results_generic) * 1.4
if calculation_choice == "Specific":
    rects5 = ax3.bar((floor_index - bar_width + 0.1), floor_SCp_results_specific_totals_show, bar_width, alpha=opacity_full,
label=environmentaloverall_label, color=environmental_labelcolor)
    rects6 = ax3.bar((floor_index - 0.1), floor_SCp_results_specific_add, bar_width, alpha=opacity, label=environmentaladdition_label,
color=environmental_labelcolor)
    autolabel_floor_env_specific(rects5)
    autolabel_floor_env_generic(rects6)
    ylim_ax3 = max(floor_SCp_results_specific_add) * 1.4
    xlim_ax3_left = min(floor_index) - positionbar
    xlim_ax3_right = max(floor_index) + 0.5
    ax3.set_ylim(0, ylim_ax3)
    ax3.set_xlim(xlim_ax3_left, xlim_ax3_right)
    ax3.grid(b=None, axis='x')
    ax3.tick_params(axis='both', labels='10', length=0)
    ax3.set_title(label='Shadow costs per product - Floor (%)' % u"\N{euro sign}", loc='center', fontsize='12', fontweight='bold')
    ax3.set_xticks(floor_index)
    ax3.set_xticklabels(floor_labels)
    ax3.legend(prop={'size': 9}, framealpha=0.5)

ax4 = self.figure.add_subplot(3, 2, 4)
def autolabel_floor_eco_generic(rects7):
    for idx, rect in enumerate(rects7):
        height = rect.get_height()
        ax4.text(rect.get_x() + rect.get_width() / 2, height, floor_C_results_generic_totals_show[idx], fontsize='11', ha='left', va='bottom')
def autolabel_floor_eco_specific(rects8):
    for idx, rect in enumerate(rects8):
        height = rect.get_height()
        ax4.text(rect.get_x() + rect.get_width() / 2, height, floor_C_results_specific_add_percentage[idx], fontsize='10', fontstyle='italic',
ha='center', va='bottom')
if calculation_choice == "Generic":
    rects7 = ax4.bar((floor_index - (bar_width / 2)), floor_C_results_generic_totals_show, bar_width, alpha=opacity_full,
label=economic_label, color=economic_labelcolor)
    autolabel_floor_eco_generic(rects7)
    ylim_ax4 = max(floor_C_results_generic) * 1.4
if calculation_choice == "Specific":
    rects7 = ax4.bar((floor_index - bar_width + 0.1), floor_C_results_specific_totals_show, bar_width, alpha=opacity_full,
label=economicoverall_label, color=economic_labelcolor)
    rects8 = ax4.bar((floor_index - 0.1), floor_C_results_specific_add, bar_width, alpha=opacity, label=economicaddition_label,
color=economic_labelcolor)
    autolabel_floor_eco_specific(rects7)
    autolabel_floor_eco_generic(rects8)
    ylim_ax4 = max(floor_C_results_specific_add) * 1.4
    xlim_ax4_left = min(floor_index) - positionbar
    xlim_ax4_right = max(floor_index) + 0.5
    ax4.set_ylim(0, ylim_ax4)
    ax4.set_xlim(xlim_ax4_left, xlim_ax4_right)
    ax4.grid(b=None, axis='x')
    ax4.tick_params(axis='both', labels='10', length=0)
    ax4.set_title(label='Costs per product - Floor (%)' % u"\N{euro sign}", loc='center', fontsize='12', fontweight='bold')
    ax4.set_xticks(floor_index)
    ax4.set_xticklabels(floor_labels)
    ax4.legend(prop={'size': 9}, framealpha=0.5)

ax5 = self.figure.add_subplot(3, 2, 5)
def autolabel_extwall_generic(rects9):
    for idx, rect in enumerate(rects9):
        height = rect.get_height()
        ax5.text(rect.get_x() + rect.get_width() / 2, height, exteriorwall_SCp_results_generic_totals_show[idx], fontsize='11', ha='left',
va='bottom')
def autolabel_extwall_specific(rects10):
    for idx, rect in enumerate(rects10):
        height = rect.get_height()
        ax5.text(rect.get_x() + rect.get_width() / 2, height, exteriorwall_SCp_results_specific_add_percentage[idx], fontsize='11',
fontstyle='italic', ha='center', va='bottom')
if calculation_choice == "Generic":
    rects9 = ax5.bar((exteriorwall_index - (bar_width / 2)), exteriorwall_SCp_results_generic_totals_show, bar_width, alpha=opacity_full,

```

```

label=environmental_label, color=environmental_labelcolor)
autolabel_extwall_generic(rects9)
ylim_ax5 = max(exteriorwall_SCp_results_generic) * 1.4
if calculation_choice == "Specific":
    rects9 = ax5.bar((exteriorwall_index - bar_width + 0.1), exteriorwall_SCp_results_specific_totals_show, bar_width, alpha=opacity_full,
label=environmentaloverall_label, color=environmental_labelcolor)
    rects10 = ax5.bar((exteriorwall_index - 0.1), exteriorwall_SCp_results_specific_add, bar_width, alpha=opacity,
label=environmentaladdition_label, color=environmental_labelcolor)
    autolabel_extwall_specific(rects9)
    autolabel_extwall_generic(rects10)
    ylim_ax5 = max(exteriorwall_SCp_results_specific_add) * 1.4
    xlim_ax5_left = min(exteriorwall_index) - positionbar
    xlim_ax5_right = max(exteriorwall_index) + 0.5
    ax5.set_ylim(0, ylim_ax5)
    ax5.set_xlim(xlim_ax5_left, xlim_ax5_right)
    ax5.grid(b=None, axis='x')
    ax5.tick_params(axis='both', labelsize='10', length=0)
    ax5.set_title(label='Shadow costs per product - Exterior Wall (%)' % u"\N{euro sign}", loc='center', fontsize='12', fontweight='bold')
    ax5.set_xticks(exteriorwall_index)
    ax5.set_xticklabels(exteriorwall_labels)
    ax5.legend(prop={'size': 9}, framealpha=0.5)

ax6 = self.figure.add_subplot(3, 2, 6)
def autolabel_extwall_eco_generic(rects11):
    for idx, rect in enumerate(rects11):
        height = rect.get_height()
        ax6.text(rect.get_x() + rect.get_width() / 2, height, exteriorwall_C_results_generic_totals_show[idx], fontsize='11', ha='left', va='bottom')
def autolabel_extwall_eco_specific(rects12):
    for idx, rect in enumerate(rects12):
        height = rect.get_height()
        ax6.text(rect.get_x() + rect.get_width() / 2, height, exteriorwall_C_results_specific_add_percentage[idx], fontsize='10', fontstyle='italic',
ha='center', va='bottom')
if calculation_choice == "Generic":
    rects11 = ax6.bar((exteriorwall_index - (bar_width / 2)), exteriorwall_C_results_generic_totals_show, bar_width, alpha=opacity_full,
label=economic_label, color=economic_labelcolor)
    autolabel_extwall_eco_generic(rects11)
    ylim_ax6 = max(exteriorwall_C_results_generic) * 1.4
if calculation_choice == "Specific":
    rects11 = ax6.bar((exteriorwall_index - bar_width + 0.1), exteriorwall_C_results_specific_totals_show, bar_width, alpha=opacity_full,
label=economicoverall_label, color=economic_labelcolor)
    rects12 = ax6.bar((exteriorwall_index - 0.1), exteriorwall_C_results_specific_add, bar_width, alpha=opacity,
label=economicaddition_label, color=economic_labelcolor)
    autolabel_extwall_eco_specific(rects11)
    autolabel_extwall_eco_generic(rects12)
    ylim_ax6 = max(exteriorwall_C_results_specific_add) * 1.4
    xlim_ax6_left = min(exteriorwall_index) - positionbar
    xlim_ax6_right = max(exteriorwall_index) + 0.5
    ax6.set_ylim(0, ylim_ax6)
    ax6.set_xlim(xlim_ax6_left, xlim_ax6_right)
    ax6.grid(b=None, axis='x')
    ax6.tick_params(axis='both', labelsize='10', length=0)
    ax6.set_title(label='Costs per product - Exterior Wall (%)' % u"\N{euro sign}", loc='center', fontsize='12', fontweight='bold')
    ax6.set_xticks(exteriorwall_index)
    ax6.set_xticklabels(exteriorwall_labels)
    ax6.legend(prop={'size': 9}, framealpha=0.5)

plt.tight_layout()
self.canvas1.draw()

# Configures the 'IFC-functionality'.
def open_IFC(self):
    ifc_path = QtGui.QFileDialog.getOpenFileName(self, 'Open IFC-file', '.', 'Industry Foundation Classes (*.ifc)')
    self.file_path = ifc_path
    ifc_file = ifcopenshell.open(str(self.file_path))
    self.ui.labelIFCFilePath.setText(os.path.basename(self.file_path))
    if self.ui.labelIFCFilePath != "":
        walls = ifc_file.by_type("IfcWall")
        floors = ifc_file.by_type("IfcSlab")
        widths = []
        areas = []
        lengths = []

```

```

heights = []
volumes = []
codes = []
element = []
material_reference = []

for wall in walls:
    for rel in wall.HasAssociations:
        if rel.is_a("IfcRelAssociatesClassification"):
            codes.append(rel.RelatingClassification.ItemReference)
            element.append("Wall")
    if wall.IsDefinedBy:
        definitions = wall.IsDefinedBy
        for definition in definitions:
            if 'IfcRelDefinesByProperties' == definition.is_a():
                property_definition = definition.RelatingPropertyDefinition
                if 'IfcElementQuantity' == property_definition.is_a():
                    for quantity in property_definition.Quantities:
                        if 'IfcQuantityLength' == quantity.is_a():
                            if quantity.Name == 'Width':
                                widths.append(quantity.LengthValue)
                            if quantity.Name == 'Height':
                                heights.append(quantity.LengthValue)
                            if quantity.Name == 'Length':
                                lengths.append(quantity.LengthValue)
                        if 'IfcQuantityArea' == quantity.is_a():
                            if quantity.Name == 'GrossSideArea':
                                areas.append(quantity.AreaValue)
                        if 'IfcQuantityVolume' == quantity.is_a():
                            if quantity.Name == 'GrossVolume':
                                volumes.append(quantity.VolumeValue)
            if 'IfcPropertySet' == property_definition.is_a():
                for reference in property_definition.HasProperties:
                    if 'IfcPropertySingleValue' == reference.is_a():
                        if reference.Name == 'Reference':
                            material_reference.append(reference.NominalValue.wrappedValue)
            if property_definition.is_a() != 'IfcElementQuantity':
                for code in property_definition.HasProperties:
                    if code.Name == 'Itemcode':
                        codes.append(code.NominalValue.wrappedValue)
                        element.append("Wall")
                    if code.Name == 'name':
                        material_reference.append(code.NominalValue.wrappedValue)
            for quantity in property_definition.HasProperties:
                if 'IfcPropertySingleValue' == quantity.is_a():
                    if quantity.Name == 'width':
                        widths.append(quantity.NominalValue.wrappedValue)
                        widths = [int(i) for i in widths]
                    if quantity.Name == 'length':
                        lengths.append(quantity.NominalValue.wrappedValue)
                        lengths = [int(i) for i in lengths]
                    if quantity.Name == 'height':
                        heights.append(quantity.NominalValue.wrappedValue)
                        heights = [int(i) for i in heights]

for floor in floors:
    for rel in floor.HasAssociations:
        if rel.is_a("IfcRelAssociatesClassification"):
            codes.append(rel.RelatingClassification.ItemReference)
            element.append("Floor")
    if floor.IsDefinedBy:
        definitions = floor.IsDefinedBy
        for definition in definitions:
            if 'IfcRelDefinesByProperties' == definition.is_a():
                property_definition = definition.RelatingPropertyDefinition
                if 'IfcElementQuantity' == property_definition.is_a():
                    for quantity in property_definition.Quantities:
                        if 'IfcQuantityLength' == quantity.is_a():
                            if quantity.Name == 'Width':
                                widths.append(quantity.LengthValue)
                            if quantity.Name == 'Height':

```

```

        heights.append(quantity.LengthValue)
        if quantity.Name == 'Length':
            lengths.append(quantity.LengthValue)
        if 'IfcQuantityArea' == quantity.is_a():
            if quantity.Name == 'GrossArea':
                areas.append(quantity.AreaValue)
        if 'IfcQuantityVolume' == quantity.is_a():
            if quantity.Name == 'GrossVolume':
                volumes.append(quantity.VolumeValue)
        if 'IfcPropertySet' == property_definition.is_a():
            for reference in property_definition.HasProperties:
                if 'IfcPropertySingleValue' == reference.is_a():
                    if reference.Name == 'Reference':
                        material_reference.append(reference.NominalValue.wrappedValue)
        if property_definition.is_a() != 'IfcElementQuantity':
            for code in property_definition.HasProperties:
                if code.Name == 'itemcode':
                    codes.append(code.NominalValue.wrappedValue)
            element.append("Floor")
            if code.Name == 'name':
                material_reference.append(code.NominalValue.wrappedValue)
        for quantity in property_definition.HasProperties:
            if 'IfcPropertySingleValue' == quantity.is_a():
                if quantity.Name == 'width':
                    widths.append(quantity.NominalValue.wrappedValue)
                    widths = [int(i) for i in widths]
                if quantity.Name == 'length':
                    lengths.append(quantity.NominalValue.wrappedValue)
                    lengths = [int(i) for i in lengths]
                if quantity.Name == 'height':
                    heights.append(quantity.NominalValue.wrappedValue)
                    heights = [int(i) for i in heights]
        if areas == []:
            lengths = np.multiply(lengths, 0.001)
            heights = np.multiply(heights, 0.001)
            areas = np.multiply(lengths, heights)

# Content of the IFC export to .xlsx
material_reference = [x[-4:] for x in material_reference]
material_reference = ["-" if "." in x else x for x in material_reference]
df = pd.DataFrame()
df['Assembly Code'] = codes
df['Element'] = element
df['Materialreference'] = material_reference
df['Area'] = areas
df['Width'] = widths
df.sort_values(by=['Assembly Code'], inplace=True)

time = datetime.datetime.now()
output_time_date = time.strftime("%d-%m-%Y %H-%M-%S")
output_time = time.strftime("%d-%m-%Y %H-%M-%S")
overviewsheet = "Current values"

# Read or export IFC to .xlsx
if os.path.isfile('Surface_ifc.xlsx'):
    workbook = openpyxl.load_workbook('Surface_ifc.xlsx')
    revision_sheet = workbook["Revisions"]
    revision_sheet.append([output_time_date, 'Checker1'])
    std = workbook.get_sheet_by_name(overviewsheet)
    workbook.remove_sheet(std)
    workbook.save("Surface_ifc.xlsx")
    workbook.close()

filename = 'Surface_ifc.xlsx'
with pd.ExcelWriter(filename, engine='openpyxl') as writer:
    # open existing excel file
    writer.book = openpyxl.load_workbook(filename)
    # add data
    df.to_excel(writer, index=False, sheet_name=output_time)
    df.to_excel(writer, index=False, sheet_name=overviewsheet)

```



```

workbook = openpyxl.load_workbook('Surface_ifc.xlsx')
workbook.active = workbook[overviewsheet]
workbook.save("Surface_ifc.xlsx")
workbook.close()
else:
    writer = pd.ExcelWriter('Surface_ifc.xlsx', engine='xlsxwriter')
    df.to_excel(writer, index=False, sheet_name=overviewsheet)
    workbook = writer.book
    worksheet = writer.sheets[overviewsheet]
    worksheet.set_column('A:A', 14)
    worksheet.set_column('B:B', 11)
    worksheet.set_column('C:C', 18)
    worksheet.set_column('D:E', 11)
    df1 = pd.DataFrame()
    df1['List of Revisions'] = [output_time_date]
    df1['Checker'] = ""
    df1['Comments'] = ""
    df1.to_excel(writer, index=False, sheet_name='Revisions')
    worksheet1 = writer.sheets['Revisions']
    worksheet1.set_column('A:A', 18)
    worksheet1.set_column('B:C', 11)
    df.to_excel(writer, index=False, sheet_name=output_time)
    worksheet = writer.sheets[output_time]
    worksheet.set_column('A:A', 14)
    worksheet.set_column('B:B', 11)
    worksheet.set_column('C:C', 18)
    worksheet.set_column('D:E', 11)
    writer.save()
    workbook = openpyxl.load_workbook('Surface_ifc.xlsx')
    workbook.active = workbook[overviewsheet]
    workbook.save("Surface_ifc.xlsx")
    workbook.close()

```

Generates the slider value change.

```

def slider_change(self):
    my_value = str(self.ui.slider.value())
    self.slideredit.setText(my_value)

```

Stores the plots in a .png or .pdf file.

```

def save_plots(self):
    formats = "Portable Network Graphics (*.png);;Adobe Acrobat (*.pdf)"
    output_file = QtGui.QFileDialog.getSaveFileName(self, 'Save Plots', 'figures/', formats)
    if output_file == "":
        return
    else:
        self.figure.savefig(output_file)

```

Closes the application.

```

def close_application(self):
    choice = QtGui.QMessageBox.question(self, "Close the application", "Are you sure you want to close the application?",
        QtGui.QMessageBox.Yes | QtGui.QMessageBox.Save | QtGui.QMessageBox.No)
    if choice == QtGui.QMessageBox.Yes:
        sys.exit()
    if choice == QtGui.QMessageBox.Save:
        formats = "Portable Network Graphics (*.png);;Adobe Acrobat (*.pdf)"
        output_file = QtGui.QFileDialog.getSaveFileName(self, 'Save Plots', 'figures/', formats)
        if output_file == "":
            return
        else:
            self.figure.savefig(output_file)
    else:
        pass

```

```

if __name__ == "__main__":
    app = QtGui.QApplication(sys.argv)
    gui = Main()
    gui.showMaximized()
    sys.exit(app.exec_())

```

FORMAT FOR A MATERIAL COMPONENT – FORMATCOMPONENT.PY

```

import pandas as pd
import numpy as np
from basevalues import *
from pathlib import Path
import os

full_path = os.path.realpath(__file__)
path, filename = os.path.split(full_path)
Type = filename.replace('.py', '')
Type = Type.replace('EC_', '')
Type = Type.replace('0', '.0')
if ".0." in Type:
    Type = Type.replace('.0.', '.0')

##### For future resesarch, materialreference addition to the surface .xlsx-sheets. #####
Reference = Type[:-4]
Reference_material = ""
Reference_type = ""
Costreference = "mcm_mean"

##### INPUT #####
product = pd.read_excel("Database.xlsx", sheet_name="product")
product.index = product["building_product"]
ac = pd.read_excel("Database.xlsx", sheet_name="ac")
ac.index = ac["ac"]
components = pd.read_excel("Database.xlsx", sheet_name="components")
components.index = components["building_product"]
pc = pd.read_excel("Database.xlsx", sheet_name="pc")
pc.index = pc["pc"]
tc = pd.read_excel("Database.xlsx", sheet_name="tc")
tc.index = tc["tc"]
cost = pd.read_excel("Database.xlsx", sheet_name="cost")
cost.index = cost["building_product"]
Cost = cost.loc[Type, Costreference]

LDp = product.loc[Type, "LDp"]
LDo = product.loc[Type, "LDo"]
Tb = product.loc[Type, 'Tb']
Dim1_def = product.loc[Type, "Dim1_def"]
Dim2_def = product.loc[Type, "Dim2_def"]
Dim1_sch = product.loc[Type, "Dim1_sch"]
Dim2_sch = product.loc[Type, "Dim2_sch"]
Dim2_b = 0
Scaling = product.loc[Type, "scaling"]
Sfo = product.loc[Type, "Sfo"]
BUTC = product.loc[Type, "base_unit_tc_component"]

##### SURFACES #####
pathsurfaceifc = Path("./Surface_ifc.xlsx")
pathsurface = Path("./Surface_generic.xlsx")
if pathsurfaceifc.is_file():
    surface = pd.read_excel("Surface_ifc.xlsx", sheet_name="Current values")
    surface.index = surface["Assembly Code"]
    E_Reference = Reference
    ### 1 ###
    if E_Reference in surface.index:
        if Reference_material != "":
            surface.index = surface["Materialreference"]
            Dim1_b_g = surface.loc[Reference_material, "Width"]
            surface_area_g = surface.loc[Reference_material, "Area"]
        elif Reference_material == "":
            surface.index = surface["Assembly Code"]
            Dim1_b_g = surface.loc[E_Reference, "Width"]
            surface_area_g = surface.loc[E_Reference, "Area"]
        else:
            Dim1_b_g = 0
            surface_area_g = 0
    ### 2 ###
    surface.index = surface["Assembly Code"]
    if Type in surface.index:

```

```

if Reference_type != "":
    surface.index = surface["Materialreference"]
    Dim1_b_s = surface.loc[Reference_type, "Width"]
    surface_area_s = surface.loc[Reference_type, "Area"]
elif Reference_type == "":
    Dim1_b_s = surface.loc[Type, "Width"]
    surface_area_s = surface.loc[Type, "Area"]
else:
    Dim1_b_s = 0
    surface_area_s = 0

countofelement_s = list(surface.index).count(Type)
countofelement_g = list(surface.index).count(Reference)

elif pathsurface.is_file():
    surface = pd.read_excel("Surface_generic.xlsx", sheet_name=0)
    surface.index = surface["Assembly Code"]
    E_Reference = float(Reference)
    countofelement_s = 1
    countofelement_g = 1
    ### 1 ###
    if E_Reference in surface.index:
        if Reference_material != "":
            surface.index = surface["Materialreference"]
            Dim1_b_g = surface.loc[Reference_material, "Width"]
            surface_area_g = surface.loc[Reference_material, "Area"]
        elif Reference_material == "":
            surface.index = surface["Assembly Code"]
            Dim1_b_g = surface.loc[E_Reference, "Width"]
            surface_area_g = surface.loc[E_Reference, "Area"]
        else:
            Dim1_b_g = 0
            surface_area_g = 0

    ### 2 ###
    surface.index = surface["Assembly Code"]
    if Type in surface.index:
        if Reference_type != "":
            surface.index = surface["Materialreference"]
            Dim1_b_s = surface.loc[Reference_type, "Width"]
            surface_area_s = surface.loc[Reference_type, "Area"]
        elif Reference_type == "":
            Dim1_b_s = surface.loc[Type, "Width"]
            surface_area_s = surface.loc[Type, "Area"]
        else:
            Dim1_b_s = 0
            surface_area_s = 0

    ### In the case that the element is (not) chosen, Dim1_b will take the first available Dim1_b
    if countofelement_s == 0:
        surface_area_spe = 0
        Dim1_b_spe = 0
    elif countofelement_s == 1:
        surface_area_spe = surface_area_s
        Dim1_b_spe = Dim1_b_s
    else:
        surface_area_spe = sum(surface_area_s)
        Dim1_b_spe = max(Dim1_b_s)

    if countofelement_g == 0:
        surface_area_gen = 0
        Dim1_b_gen = 0
    elif countofelement_g == 1:
        surface_area_gen = surface_area_g
        Dim1_b_gen = Dim1_b_g
    else:
        surface_area_gen = sum(surface_area_g)
        Dim1_b_gen = max(Dim1_b_g)

    if Type in surface.index:
        Dim1_b = Dim1_b_spe

```

```

elif E_Reference in surface.index:
    Dim1_b = Dim1_b_gen

##### CALCULATIONS #####
LDreken = min(LDp, LDg)
Vast1 = Dim1_def - Dim1_sch
Vast2 = Dim2_def - Dim2_sch
Sfp_1 = Dim1_b - Vast1
Sfp_2 = Dim1_def - Vast1
Sfp_3 = Dim2_b - Vast2
Sfp_4 = Dim2_def - Vast2

if Dim2_def or Vast1 or Vast2 == 0:
    if Sfp_2 == 0:
        Sfp = 0
    else:
        Sfp = (Sfp_1/Sfp_2)
    else:
        Sfp = (Sfp_1/Sfp_2) * (Sfp_3/Sfp_4)
if Scaling == "N.v.t.":
    Sftot = 1
elif Scaling == "Massa volgens tabel":
    Sftot = 1
elif Scaling == "Rechtevenredig":
    Sftot = Sfp * Sfo
LDf = np.maximum(1, float(LDreken / LDo))

### FOR DETERMINING NUMBER OF COMPONENTS IN ELEMENT ###
usedcomponents = components.loc[Type, "building_product"]
try:
    listcomponents = list(usedcomponents[Type])
    numberofcomponents = listcomponents.count(Type)
except:
    numberofcomponents = 1

components.index = components["component_nr"]
if numberofcomponents > 0:
    Type_1 = (Type + "-1")
    pc_1 = components.loc[Type_1, 'pc']
    ac_1 = components.loc[Type_1, 'ac']
    hvh_1 = components.loc[Type_1, 'hvh']
    Bafv_1 = components.loc[Type_1, 'Bafv']
    fs_1 = components.loc[Type_1, 'fs']
    fv_1 = components.loc[Type_1, 'fv']
    fr_1 = components.loc[Type_1, 'fr']
    fh_1 = components.loc[Type_1, 'fh']
    fe_1 = components.loc[Type_1, 'fe']
    BUPC_1 = components.loc[Type_1, "base_unit_pc_component"]
    BUAC_1 = components.loc[Type_1, "base_unit_ac_component"]

if ac_1 == "00" or BUPC_1 != "kg":
    BMEF_ac_1 = 0
else:
    ac_1_ADPcs = ac.loc[ac_1, "ADPcs"]
    ac_1_ADPfs = ac.loc[ac_1, "ADPfs"]
    ac_1_GWPs = ac.loc[ac_1, "GWPs"]
    ac_1_ODPs = ac.loc[ac_1, "ODPs"]
    ac_1_POCPs = ac.loc[ac_1, "POCPs"]
    ac_1_APs = ac.loc[ac_1, "APs"]
    ac_1_EPs = ac.loc[ac_1, "EPs"]
    ac_1_HTPs = ac.loc[ac_1, "HTPs"]
    ac_1_FAETPs = ac.loc[ac_1, "FAETPs"]
    ac_1_MAETPs = ac.loc[ac_1, "MAETPs"]
    ac_1_TETPs = ac.loc[ac_1, "TETPs"]
    ac_1_ADPcv = ac.loc[ac_1, "ADPcv"]
    ac_1_ADPfv = ac.loc[ac_1, "ADPfv"]
    ac_1_GWPv = ac.loc[ac_1, "GWPv"]
    ac_1_ODPv = ac.loc[ac_1, "ODPv"]
    ac_1_POCPv = ac.loc[ac_1, "POCPv"]
    ac_1_APv = ac.loc[ac_1, "APv"]
    ac_1_EPv = ac.loc[ac_1, "EPv"]

```

```

ac_1_HTPv = ac.loc[ac_1, "HTPv"]
ac_1_FAETPv = ac.loc[ac_1, "FAETPv"]
ac_1_MAETPv = ac.loc[ac_1, "MAETPv"]
ac_1_TETPv = ac.loc[ac_1, "TETPv"]
ac_1_ADPr = ac.loc[ac_1, "ADPr"]
ac_1_ADPr = ac.loc[ac_1, "ADPr"]
ac_1_GWPr = ac.loc[ac_1, "GWPr"]
ac_1_ODPr = ac.loc[ac_1, "ODPr"]
ac_1_POCPv = ac.loc[ac_1, "POCPv"]
ac_1_APr = ac.loc[ac_1, "APr"]
ac_1_EPr = ac.loc[ac_1, "EPr"]
ac_1_HTPr = ac.loc[ac_1, "HTPr"]
ac_1_FAETPr = ac.loc[ac_1, "FAETPr"]
ac_1_MAETPr = ac.loc[ac_1, "MAETPr"]
ac_1_TETPr = ac.loc[ac_1, "TETPr"]
ac_1_ADPh = ac.loc[ac_1, "ADPh"]
ac_1_ADPh = ac.loc[ac_1, "ADPh"]
ac_1_GWPh = ac.loc[ac_1, "GWPh"]
ac_1_ODPh = ac.loc[ac_1, "ODPh"]
ac_1_POCPv = ac.loc[ac_1, "POCPv"]
ac_1_APh = ac.loc[ac_1, "APh"]
ac_1_EPh = ac.loc[ac_1, "EPh"]
ac_1_HTPh = ac.loc[ac_1, "HTPh"]
ac_1_FAETPh = ac.loc[ac_1, "FAETPh"]
ac_1_MAETPh = ac.loc[ac_1, "MAETPh"]
ac_1_TETPh = ac.loc[ac_1, "TETPh"]
ac_1_ADPe = ac.loc[ac_1, "ADPe"]
ac_1_ADPe = ac.loc[ac_1, "ADPe"]
ac_1_GWPe = ac.loc[ac_1, "GWPe"]
ac_1_ODPe = ac.loc[ac_1, "ODPe"]
ac_1_POCPe = ac.loc[ac_1, "POCPe"]
ac_1_APe = ac.loc[ac_1, "APe"]
ac_1_EPe = ac.loc[ac_1, "EPe"]
ac_1_HTPe = ac.loc[ac_1, "HTPe"]
ac_1_FAETPe = ac.loc[ac_1, "FAETPe"]
ac_1_MAETPe = ac.loc[ac_1, "MAETPe"]
ac_1_TETPe = ac.loc[ac_1, "TETPe"]
BMEF_pc_1_ADPr = pc.loc[pc_1, "ADPr"]
BMEF_pc_1_ADPr = pc.loc[pc_1, "ADPr"]
BMEF_pc_1_GWPr = pc.loc[pc_1, "GWPr"]
BMEF_pc_1_ODPr = pc.loc[pc_1, "ODPr"]
BMEF_pc_1_POCPv = pc.loc[pc_1, "POCPv"]
BMEF_pc_1_APr = pc.loc[pc_1, "APr"]
BMEF_pc_1_EPr = pc.loc[pc_1, "EPr"]
BMEF_pc_1_HTPr = pc.loc[pc_1, "HTPr"]
BMEF_pc_1_FAETPr = pc.loc[pc_1, "FAETPr"]
BMEF_pc_1_MAETPr = pc.loc[pc_1, "MAETPr"]
BMEF_pc_1_TETPr = pc.loc[pc_1, "TETPr"]
BMEF_ac_1_ADPr = float(fs_1 * (ac_1_ADPrs + tc["ADPr"] / 1000 * fTvol * Ts) + fv_1 * (
    ac_1_ADPrv + tc["ADPr"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_ADPr + tc["ADPr"] / 1000 * fTvol * Tr) + fh_1 * (
    ac_1_ADPh + tc["ADPr"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_ADPe + tc["ADPr"] / 1000 * fTvol * Te))
BMEF_ac_1_ADPr = float(fs_1 * (ac_1_ADPrs + tc["ADPr"] / 1000 * fTvol * Ts) + fv_1 * (
    ac_1_ADPrv + tc["ADPr"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_ADPr + tc["ADPr"] / 1000 * fTvol * Tr) + fh_1 * (
    ac_1_ADPh + tc["ADPr"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_ADPe + tc["ADPr"] / 1000 * fTvol * Te))
BMEF_ac_1_GWPr = float(
    fs_1 * (ac_1_GWPrs + tc["GWPr"] / 1000 * fTvol * Ts) + fv_1 * (ac_1_GWPrv + tc["GWPr"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_GWPr + tc["GWPr"] / 1000 * fTvol * Tr) + fh_1 * (ac_1_GWPh + tc["GWPr"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_GWPe + tc["GWPr"] / 1000 * fTvol * Te))
BMEF_ac_1_ODPr = float(
    fs_1 * (ac_1_ODPrs + tc["ODPr"] / 1000 * fTvol * Ts) + fv_1 * (ac_1_ODPrv + tc["ODPr"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_ODPr + tc["ODPr"] / 1000 * fTvol * Tr) + fh_1 * (ac_1_ODPh + tc["ODPr"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_ODPe + tc["ODPr"] / 1000 * fTvol * Te))
BMEF_ac_1_POCPv = float(fs_1 * (ac_1_POCPs + tc["POCPv"] / 1000 * fTvol * Ts) + fv_1 * (
    ac_1_POCPv + tc["POCPv"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_POCP + tc["POCPv"] / 1000 * fTvol * Tr) + fh_1 * (
    ac_1_POCPv + tc["POCPv"] / 1000 * fTvol * Th) + \

```

```

    fe_1 * (ac_1_POCPe + tc["POCP"] / 1000 * fTvol * Te))
BMEF_ac_1_AP = float(
    fs_1 * (ac_1_APs + tc["AP"] / 1000 * fTvol * Ts) + fv_1 * (ac_1_APv + tc["AP"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_APr + tc["AP"] / 1000 * fTvol * Tr) + fh_1 * (ac_1_APh + tc["AP"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_APe + tc["AP"] / 1000 * fTvol * Te))
BMEF_ac_1_EP = float(
    fs_1 * (ac_1_EPs + tc["EP"] / 1000 * fTvol * Ts) + fv_1 * (ac_1_Epv + tc["EP"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_EPr + tc["EP"] / 1000 * fTvol * Tr) + fh_1 * (ac_1_EPh + tc["EP"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_EPe + tc["EP"] / 1000 * fTvol * Te))
BMEF_ac_1_HTP = float(
    fs_1 * (ac_1_HTPs + tc["HTP"] / 1000 * fTvol * Ts) + fv_1 * (ac_1_HTPv + tc["HTP"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_HTPr + tc["HTP"] / 1000 * fTvol * Tr) + fh_1 * (ac_1_HTPh + tc["HTP"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_HTPe + tc["HTP"] / 1000 * fTvol * Te))
BMEF_ac_1_FAETP = float(fs_1 * (ac_1_FAETPs + tc["FAETP"] / 1000 * fTvol * Ts) + fv_1 * (
    ac_1_FAETpv + tc["FAETP"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_FAETPr + tc["FAETP"] / 1000 * fTvol * Tr) + fh_1 * (
    ac_1_FAETPh + tc["FAETP"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_FAETPe + tc["FAETP"] / 1000 * fTvol * Te))
BMEF_ac_1_MAETP = float(fs_1 * (ac_1_MAETPs + tc["MAETP"] / 1000 * fTvol * Ts) + fv_1 * (
    ac_1_MAETpv + tc["MAETP"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_MAETPr + tc["MAETP"] / 1000 * fTvol * Tr) + fh_1 * (
    ac_1_MAETPh + tc["MAETP"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_MAETPe + tc["MAETP"] / 1000 * fTvol * Te))
BMEF_ac_1_TETP = float(fs_1 * (ac_1_TETPs + tc["TETP"] / 1000 * fTvol * Ts) + fv_1 * (
    ac_1_TETpv + tc["TETP"] / 1000 * fTvol * Tv) + \
    fr_1 * (ac_1_TETPr + tc["TETP"] / 1000 * fTvol * Tr) + fh_1 * (
    ac_1_TETPh + tc["TETP"] / 1000 * fTvol * Th) + \
    fe_1 * (ac_1_TETPe + tc["TETP"] / 1000 * fTvol * Te))

MEFeh_ADPC_1 = float((1 + Ogen) * (BMEF_pc_1_ADPC + tc["ADPC"] / 1000 * Tb * fTvol + BMEF_ac_1_ADPC))
MEFeh_ADPF_1 = float((1 + Ogen) * (BMEF_pc_1_ADPF + tc["ADPF"] / 1000 * Tb * fTvol + BMEF_ac_1_ADPF))
MEFeh_GWP_1 = float((1 + Ogen) * (BMEF_pc_1_GWP + tc["GWP"] / 1000 * Tb * fTvol + BMEF_ac_1_GWP))
MEFeh_ODP_1 = float((1 + Ogen) * (BMEF_pc_1_ODP + tc["ODP"] / 1000 * Tb * fTvol + BMEF_ac_1_ODP))
MEFeh_POCP_1 = float((1 + Ogen) * (BMEF_pc_1_POCP + tc["POCP"] / 1000 * Tb * fTvol + BMEF_ac_1_POCP))
MEFeh_AP_1 = float((1 + Ogen) * (BMEF_pc_1_AP + tc["AP"] / 1000 * Tb * fTvol + BMEF_ac_1_AP))
MEFeh_EP_1 = float((1 + Ogen) * (BMEF_pc_1_EP + tc["EP"] / 1000 * Tb * fTvol + BMEF_ac_1_EP))
MEFeh_HTP_1 = float((1 + Ogen) * (BMEF_pc_1_HTP + tc["HTP"] / 1000 * Tb * fTvol + BMEF_ac_1_HTP))
MEFeh_FAETP_1 = float((1 + Ogen) * (BMEF_pc_1_FAETP + tc["FAETP"] / 1000 * Tb * fTvol + BMEF_ac_1_FAETP))
MEFeh_MAETP_1 = float((1 + Ogen) * (BMEF_pc_1_MAETP + tc["MAETP"] / 1000 * Tb * fTvol + BMEF_ac_1_MAETP))
MEFeh_TETP_1 = float((1 + Ogen) * (BMEF_pc_1_TETP + tc["TETP"] / 1000 * Tb * fTvol + BMEF_ac_1_TETP))

MEFo_1_ADPC = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_ADPC_1
MEFo_1_ADPF = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_ADPF_1
MEFo_1_GWP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_GWP_1
MEFo_1_ODP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_ODP_1
MEFo_1_POCP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_POCP_1
MEFo_1_AP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_AP_1
MEFo_1_EP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_EP_1
MEFo_1_HTP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_HTP_1
MEFo_1_FAETP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_FAETP_1
MEFo_1_MAETP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_MAETP_1
MEFo_1_TETP = hvh_1 * Sftot * (1 + Bafv_1) * LDf * MEFeh_TETP_1

MEFo_ADPC = MEFo_1_ADPC
MEFo_ADPF = MEFo_1_ADPF
MEFo_GWP = MEFo_1_GWP
MEFo_ODP = MEFo_1_ODP
MEFo_POCP = MEFo_1_POCP
MEFo_AP = MEFo_1_AP
MEFo_EP = MEFo_1_EP
MEFo_HTP = MEFo_1_HTP
MEFo_FAETP = MEFo_1_FAETP
MEFo_MAETP = MEFo_1_MAETP
MEFo_TETP = MEFo_1_TETP

if numberofcomponents > 1:
    Type_2 = (Type + "-2")
    pc_2 = components.loc[Type_2, 'pc']
    ac_2 = components.loc[Type_2, 'ac']
    hvh_2 = components.loc[Type_2, 'hvh']
    Bafv_2 = components.loc[Type_2, 'Bafv']

```



```

fs_2 = components.loc[Type_2, 'fs']
fv_2 = components.loc[Type_2, 'fv']
fr_2 = components.loc[Type_2, 'fr']
fh_2 = components.loc[Type_2, 'fh']
fe_2 = components.loc[Type_2, 'fe']
BUPC_2 = components.loc[Type_2, "base_unit_pc_component"]
BUAC_2 = components.loc[Type_2, "base_unit_ac_component"]

```

```

if ac_2 == "00" or BUPC_2 != "kg":
    BMEF_ac_2 = 0
else:
    ac_2_ADPCs = ac.loc[ac_2, "ADPCs"]
    ac_2_ADPFs = ac.loc[ac_2, "ADPFs"]
    ac_2_GWPs = ac.loc[ac_2, "GWPs"]
    ac_2_ODPs = ac.loc[ac_2, "ODPs"]
    ac_2_POCPs = ac.loc[ac_2, "POCPs"]
    ac_2_APs = ac.loc[ac_2, "APs"]
    ac_2_EPs = ac.loc[ac_2, "EPs"]
    ac_2_HTPs = ac.loc[ac_2, "HTPs"]
    ac_2_FAETPs = ac.loc[ac_2, "FAETPs"]
    ac_2_MAETPs = ac.loc[ac_2, "MAETPs"]
    ac_2_TETPs = ac.loc[ac_2, "TETPs"]
    ac_2_ADPCv = ac.loc[ac_2, "ADPCv"]
    ac_2_ADPFv = ac.loc[ac_2, "ADPFv"]
    ac_2_GWpv = ac.loc[ac_2, "GWpv"]
    ac_2_ODpv = ac.loc[ac_2, "ODpv"]
    ac_2_POCPv = ac.loc[ac_2, "POCPv"]
    ac_2_APv = ac.loc[ac_2, "APv"]
    ac_2_Epv = ac.loc[ac_2, "EPv"]
    ac_2_HTPv = ac.loc[ac_2, "HTPv"]
    ac_2_FAETpv = ac.loc[ac_2, "FAETpv"]
    ac_2_MAETpv = ac.loc[ac_2, "MAETpv"]
    ac_2_TETpv = ac.loc[ac_2, "TETpv"]
    ac_2_ADPCr = ac.loc[ac_2, "ADPCr"]
    ac_2_ADPfr = ac.loc[ac_2, "ADPfr"]
    ac_2_GWPr = ac.loc[ac_2, "GWPr"]
    ac_2_ODPr = ac.loc[ac_2, "ODPr"]
    ac_2_POCPr = ac.loc[ac_2, "POCPr"]
    ac_2_APr = ac.loc[ac_2, "APr"]
    ac_2_EPr = ac.loc[ac_2, "EPr"]
    ac_2_HTPr = ac.loc[ac_2, "HTPr"]
    ac_2_FAETPr = ac.loc[ac_2, "FAETPr"]
    ac_2_MAETPr = ac.loc[ac_2, "MAETPr"]
    ac_2_TETPr = ac.loc[ac_2, "TETPr"]
    ac_2_ADPCch = ac.loc[ac_2, "ADPCch"]
    ac_2_ADPFch = ac.loc[ac_2, "ADPFch"]
    ac_2_GWPh = ac.loc[ac_2, "GWPh"]
    ac_2_ODPh = ac.loc[ac_2, "ODPh"]
    ac_2_POCPch = ac.loc[ac_2, "POCPch"]
    ac_2_APh = ac.loc[ac_2, "APh"]
    ac_2_EPh = ac.loc[ac_2, "EPh"]
    ac_2_HTPch = ac.loc[ac_2, "HTPh"]
    ac_2_FAETPh = ac.loc[ac_2, "FAETPh"]
    ac_2_MAETPh = ac.loc[ac_2, "MAETPh"]
    ac_2_TETPh = ac.loc[ac_2, "TETPh"]
    ac_2_ADPCe = ac.loc[ac_2, "ADPCe"]
    ac_2_ADPfe = ac.loc[ac_2, "ADPfe"]
    ac_2_GWPe = ac.loc[ac_2, "GWPe"]
    ac_2_ODPe = ac.loc[ac_2, "ODPe"]
    ac_2_POCPe = ac.loc[ac_2, "POCPe"]
    ac_2_APe = ac.loc[ac_2, "APe"]
    ac_2_EPe = ac.loc[ac_2, "EPe"]
    ac_2_HTPe = ac.loc[ac_2, "HTPe"]
    ac_2_FAETPe = ac.loc[ac_2, "FAETPe"]
    ac_2_MAETPe = ac.loc[ac_2, "MAETPe"]
    ac_2_TETPe = ac.loc[ac_2, "TETPe"]
    BMEF_pc_2_ADPC = pc.loc[pc_2, "ADPC"]
    BMEF_pc_2_ADPF = pc.loc[pc_2, "ADPF"]
    BMEF_pc_2_GWP = pc.loc[pc_2, "GWP"]
    BMEF_pc_2_ODP = pc.loc[pc_2, "ODP"]
    BMEF_pc_2_POCP = pc.loc[pc_2, "POCP"]

```

```

BMEF_pc_2_AP = pc.loc[pc_2, "AP"]
BMEF_pc_2_EP = pc.loc[pc_2, "EP"]
BMEF_pc_2_HTP = pc.loc[pc_2, "HTP"]
BMEF_pc_2_FAETP = pc.loc[pc_2, "FAETP"]
BMEF_pc_2_MAETP = pc.loc[pc_2, "MAETP"]
BMEF_pc_2_TETP = pc.loc[pc_2, "TETP"]

BMEF_ac_2_ADPC = float(fs_2 * (ac_2_ADPCs + tc["ADPC"] / 1000 * fTvol * Ts) + fv_2 * (
    ac_2_ADPCv + tc["ADPC"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_ADPCr + tc["ADPC"] / 1000 * fTvol * Tr) + fh_2 * (
    ac_2_ADPCch + tc["ADPC"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_ADPCe + tc["ADPC"] / 1000 * fTvol * Te))
BMEF_ac_2_ADPF = float(fs_2 * (ac_2_ADPFs + tc["ADPF"] / 1000 * fTvol * Ts) + fv_2 * (
    ac_2_ADPFv + tc["ADPF"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_ADPFr + tc["ADPF"] / 1000 * fTvol * Tr) + fh_2 * (
    ac_2_ADPFch + tc["ADPF"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_ADPFv + tc["ADPF"] / 1000 * fTvol * Te))
BMEF_ac_2_GWP = float(
    fs_2 * (ac_2_GWPs + tc["GWP"] / 1000 * fTvol * Ts) + fv_2 * (ac_2_GWPv + tc["GWP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_GWPr + tc["GWP"] / 1000 * fTvol * Tr) + fh_2 * (ac_2_GWPh + tc["GWP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_GWPe + tc["GWP"] / 1000 * fTvol * Te))
BMEF_ac_2_ODP = float(
    fs_2 * (ac_2_ODPs + tc["ODP"] / 1000 * fTvol * Ts) + fv_2 * (ac_2_ODPv + tc["ODP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_ODPr + tc["ODP"] / 1000 * fTvol * Tr) + fh_2 * (ac_2_ODPh + tc["ODP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_ODPe + tc["ODP"] / 1000 * fTvol * Te))
BMEF_ac_2_POCP = float(fs_2 * (ac_2_POCPs + tc["POCP"] / 1000 * fTvol * Ts) + fv_2 * (
    ac_2_POCPv + tc["POCP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_POCPv + tc["POCP"] / 1000 * fTvol * Tr) + fh_2 * (
    ac_2_POCPch + tc["POCP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_POCPe + tc["POCP"] / 1000 * fTvol * Te))
BMEF_ac_2_AP = float(
    fs_2 * (ac_2_APs + tc["AP"] / 1000 * fTvol * Ts) + fv_2 * (ac_2_APv + tc["AP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_APr + tc["AP"] / 1000 * fTvol * Tr) + fh_2 * (ac_2_APh + tc["AP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_APe + tc["AP"] / 1000 * fTvol * Te))
BMEF_ac_2_EP = float(
    fs_2 * (ac_2_EPs + tc["EP"] / 1000 * fTvol * Ts) + fv_2 * (ac_2_EPv + tc["EP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_EPr + tc["EP"] / 1000 * fTvol * Tr) + fh_2 * (ac_2_EPh + tc["EP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_EPe + tc["EP"] / 1000 * fTvol * Te))
BMEF_ac_2_HTP = float(
    fs_2 * (ac_2_HTPs + tc["HTP"] / 1000 * fTvol * Ts) + fv_2 * (ac_2_HTPv + tc["HTP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_HTPv + tc["HTP"] / 1000 * fTvol * Tr) + fh_2 * (ac_2_HTPch + tc["HTP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_HTPe + tc["HTP"] / 1000 * fTvol * Te))
BMEF_ac_2_FAETP = float(fs_2 * (ac_2_FAETPs + tc["FAETP"] / 1000 * fTvol * Ts) + fv_2 * (
    ac_2_FAETPv + tc["FAETP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_FAETPr + tc["FAETP"] / 1000 * fTvol * Tr) + fh_2 * (
    ac_2_FAETPh + tc["FAETP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_FAETPe + tc["FAETP"] / 1000 * fTvol * Te))
BMEF_ac_2_MAETP = float(fs_2 * (ac_2_MAETPs + tc["MAETP"] / 1000 * fTvol * Ts) + fv_2 * (
    ac_2_MAETPv + tc["MAETP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_MAETPr + tc["MAETP"] / 1000 * fTvol * Tr) + fh_2 * (
    ac_2_MAETPh + tc["MAETP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_MAETPe + tc["MAETP"] / 1000 * fTvol * Te))
BMEF_ac_2_TETP = float(fs_2 * (ac_2_TETPs + tc["TETP"] / 1000 * fTvol * Ts) + fv_2 * (
    ac_2_TETPv + tc["TETP"] / 1000 * fTvol * Tv) + \
    fr_2 * (ac_2_TETPr + tc["TETP"] / 1000 * fTvol * Tr) + fh_2 * (
    ac_2_TETPh + tc["TETP"] / 1000 * fTvol * Th) + \
    fe_2 * (ac_2_TETPe + tc["TETP"] / 1000 * fTvol * Te))
#####

MEFeh_ADPC_2 = float((1 + Ogen) * (BMEF_pc_2_ADPC + tc["ADPC"] / 1000 * Tb * fTvol + BMEF_ac_2_ADPC))
MEFeh_ADPF_2 = float((1 + Ogen) * (BMEF_pc_2_ADPF + tc["ADPF"] / 1000 * Tb * fTvol + BMEF_ac_2_ADPF))
MEFeh_GWP_2 = float((1 + Ogen) * (BMEF_pc_2_GWP + tc["GWP"] / 1000 * Tb * fTvol + BMEF_ac_2_GWP))
MEFeh_ODP_2 = float((1 + Ogen) * (BMEF_pc_2_ODP + tc["ODP"] / 1000 * Tb * fTvol + BMEF_ac_2_ODP))
MEFeh_POCP_2 = float((1 + Ogen) * (BMEF_pc_2_POCP + tc["POCP"] / 1000 * Tb * fTvol + BMEF_ac_2_POCP))
MEFeh_AP_2 = float((1 + Ogen) * (BMEF_pc_2_AP + tc["AP"] / 1000 * Tb * fTvol + BMEF_ac_2_AP))
MEFeh_EP_2 = float((1 + Ogen) * (BMEF_pc_2_EP + tc["EP"] / 1000 * Tb * fTvol + BMEF_ac_2_EP))
MEFeh_HTP_2 = float((1 + Ogen) * (BMEF_pc_2_HTP + tc["HTP"] / 1000 * Tb * fTvol + BMEF_ac_2_HTP))
MEFeh_FAETP_2 = float((1 + Ogen) * (BMEF_pc_2_FAETP + tc["FAETP"] / 1000 * Tb * fTvol + BMEF_ac_2_FAETP))
MEFeh_MAETP_2 = float((1 + Ogen) * (BMEF_pc_2_MAETP + tc["MAETP"] / 1000 * Tb * fTvol + BMEF_ac_2_MAETP))
MEFeh_TETP_2 = float((1 + Ogen) * (BMEF_pc_2_TETP + tc["TETP"] / 1000 * Tb * fTvol + BMEF_ac_2_TETP))

```

```

MEFo_2_ADPc = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_ADPc_2
MEFo_2_ADPf = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_ADPf_2
MEFo_2_GWP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_GWP_2
MEFo_2_ODP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_ODP_2
MEFo_2_POCP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_POCP_2
MEFo_2_AP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_AP_2
MEFo_2_EP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_EP_2
MEFo_2_HTP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_HTP_2
MEFo_2_FAETP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_FAETP_2
MEFo_2_MAETP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_MAETP_2
MEFo_2_TETP = hvh_2 * Sftot * (1 + Bafv_2) * LDf * MEfeh_TETP_2

```

```

MEFo_ADPc = MEFo_1_ADPc + MEFo_2_ADPc
MEFo_ADPf = MEFo_1_ADPf + MEFo_2_ADPf
MEFo_GWP = MEFo_1_GWP + MEFo_2_GWP
MEFo_ODP = MEFo_1_ODP + MEFo_2_ODP
MEFo_POCP = MEFo_1_POCP + MEFo_2_POCP
MEFo_AP = MEFo_1_AP + MEFo_2_AP
MEFo_EP = MEFo_1_EP + MEFo_2_EP
MEFo_HTP = MEFo_1_HTP + MEFo_2_HTP
MEFo_FAETP = MEFo_1_FAETP + MEFo_2_FAETP
MEFo_MAETP = MEFo_1_MAETP + MEFo_2_MAETP
MEFo_TETP = MEFo_1_TETP + MEFo_2_TETP

```

```

if numberofcomponents > 2:

```

```

    Type_3 = (Type + "-3")
    pc_3 = components.loc[Type_3, 'pc']
    ac_3 = components.loc[Type_3, 'ac']
    hvh_3 = components.loc[Type_3, 'hvh']
    Bafv_3 = components.loc[Type_3, 'Bafv']
    fs_3 = components.loc[Type_3, 'fs']
    fv_3 = components.loc[Type_3, 'fv']
    fr_3 = components.loc[Type_3, 'fr']
    fh_3 = components.loc[Type_3, 'fh']
    fe_3 = components.loc[Type_3, 'fe']
    BUPC_3 = components.loc[Type_3, "base_unit_pc_component"]
    BUAC_3 = components.loc[Type_3, "base_unit_ac_component"]

```

```

if ac_3 == "00" or BUPC_3 != "kg":

```

```

    BMEF_ac_3 = 0

```

```

else:

```

```

    ac_3_ADPcs = ac.loc[ac_3, "ADPcs"]
    ac_3_ADPfs = ac.loc[ac_3, "ADPfs"]
    ac_3_GWPs = ac.loc[ac_3, "GWPs"]
    ac_3_ODPs = ac.loc[ac_3, "ODPs"]
    ac_3_POCPs = ac.loc[ac_3, "POCPs"]
    ac_3_AP = ac.loc[ac_3, "APs"]
    ac_3_EP = ac.loc[ac_3, "EPs"]
    ac_3_HTPs = ac.loc[ac_3, "HTPs"]
    ac_3_FAETPs = ac.loc[ac_3, "FAETPs"]
    ac_3_MAETPs = ac.loc[ac_3, "MAETPs"]
    ac_3_TETPs = ac.loc[ac_3, "TETPs"]
    ac_3_ADPcv = ac.loc[ac_3, "ADPcv"]
    ac_3_ADPfv = ac.loc[ac_3, "ADPfv"]
    ac_3_GWpv = ac.loc[ac_3, "GWpv"]
    ac_3_ODpv = ac.loc[ac_3, "ODpv"]
    ac_3_POCPv = ac.loc[ac_3, "POCPv"]
    ac_3_APv = ac.loc[ac_3, "APv"]
    ac_3_EPv = ac.loc[ac_3, "EPv"]
    ac_3_HTPv = ac.loc[ac_3, "HTPv"]
    ac_3_FAETpv = ac.loc[ac_3, "FAETpv"]
    ac_3_MAETpv = ac.loc[ac_3, "MAETpv"]
    ac_3_TETpv = ac.loc[ac_3, "TETpv"]
    ac_3_ADPcr = ac.loc[ac_3, "ADPcr"]
    ac_3_ADPfr = ac.loc[ac_3, "ADPfr"]
    ac_3_GWPr = ac.loc[ac_3, "GWPr"]
    ac_3_ODPr = ac.loc[ac_3, "ODPr"]
    ac_3_POCPPr = ac.loc[ac_3, "POCPPr"]
    ac_3_APr = ac.loc[ac_3, "APr"]
    ac_3_EPr = ac.loc[ac_3, "EPr"]
    ac_3_HTPPr = ac.loc[ac_3, "HTPPr"]

```

```

ac_3_FAETPr = ac.loc[ac_3, "FAETPr"]
ac_3_MAETPr = ac.loc[ac_3, "MAETPr"]
ac_3_TETPr = ac.loc[ac_3, "TETPr"]
ac_3_ADPch = ac.loc[ac_3, "ADPch"]
ac_3_ADPfh = ac.loc[ac_3, "ADPfh"]
ac_3_GWPh = ac.loc[ac_3, "GWPh"]
ac_3_ODPh = ac.loc[ac_3, "ODPh"]
ac_3_POCP = ac.loc[ac_3, "POCP"]
ac_3_APh = ac.loc[ac_3, "APh"]
ac_3_EPh = ac.loc[ac_3, "EPh"]
ac_3_HTP = ac.loc[ac_3, "HTP"]
ac_3_FAETPh = ac.loc[ac_3, "FAETPh"]
ac_3_MAETPh = ac.loc[ac_3, "MAETPh"]
ac_3_TETPh = ac.loc[ac_3, "TETPh"]
ac_3_ADPce = ac.loc[ac_3, "ADPce"]
ac_3_ADPfe = ac.loc[ac_3, "ADPfe"]
ac_3_GWPe = ac.loc[ac_3, "GWPe"]
ac_3_ODPe = ac.loc[ac_3, "ODPe"]
ac_3_POCPe = ac.loc[ac_3, "POCPe"]
ac_3_APe = ac.loc[ac_3, "APe"]
ac_3_EPe = ac.loc[ac_3, "EPe"]
ac_3_HTPe = ac.loc[ac_3, "HTPe"]
ac_3_FAETPe = ac.loc[ac_3, "FAETPe"]
ac_3_MAETPe = ac.loc[ac_3, "MAETPe"]
ac_3_TETPe = ac.loc[ac_3, "TETPe"]
BMEF_pc_3_ADPc = pc.loc[pc_3, "ADPc"]
BMEF_pc_3_ADPf = pc.loc[pc_3, "ADPf"]
BMEF_pc_3_GWP = pc.loc[pc_3, "GWP"]
BMEF_pc_3_ODP = pc.loc[pc_3, "ODP"]
BMEF_pc_3_POCP = pc.loc[pc_3, "POCP"]
BMEF_pc_3_AP = pc.loc[pc_3, "AP"]
BMEF_pc_3_EP = pc.loc[pc_3, "EP"]
BMEF_pc_3_HTP = pc.loc[pc_3, "HTP"]
BMEF_pc_3_FAETP = pc.loc[pc_3, "FAETP"]
BMEF_pc_3_MAETP = pc.loc[pc_3, "MAETP"]
BMEF_pc_3_TETP = pc.loc[pc_3, "TETP"]

BMEF_ac_3_ADPc = float(fs_3 * (ac_3_ADPcs + tc["ADPc"] / 1000 * fTvol * Ts) + fv_3 * (
    ac_3_ADPcv + tc["ADPc"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_ADPcr + tc["ADPc"] / 1000 * fTvol * Tr) + fh_3 * (
    ac_3_ADPch + tc["ADPc"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_ADPce + tc["ADPc"] / 1000 * fTvol * Te))
BMEF_ac_3_ADPf = float(fs_3 * (ac_3_ADPfs + tc["ADPf"] / 1000 * fTvol * Ts) + fv_3 * (
    ac_3_ADPfv + tc["ADPf"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_ADPfr + tc["ADPf"] / 1000 * fTvol * Tr) + fh_3 * (
    ac_3_ADPfh + tc["ADPf"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_ADPfe + tc["ADPf"] / 1000 * fTvol * Te))
BMEF_ac_3_GWP = float(
    fs_3 * (ac_3_GWPs + tc["GWP"] / 1000 * fTvol * Ts) + fv_3 * (ac_3_GWpv + tc["GWP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_GWPr + tc["GWP"] / 1000 * fTvol * Tr) + fh_3 * (ac_3_GWPh + tc["GWP"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_GWPe + tc["GWP"] / 1000 * fTvol * Te))
BMEF_ac_3_ODP = float(
    fs_3 * (ac_3_ODPs + tc["ODP"] / 1000 * fTvol * Ts) + fv_3 * (ac_3_ODPv + tc["ODP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_ODPr + tc["ODP"] / 1000 * fTvol * Tr) + fh_3 * (ac_3_ODPh + tc["ODP"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_ODPe + tc["ODP"] / 1000 * fTvol * Te))
BMEF_ac_3_POCP = float(fs_3 * (ac_3_POCPs + tc["POCP"] / 1000 * fTvol * Ts) + fv_3 * (
    ac_3_POCPv + tc["POCP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_POCPPr + tc["POCP"] / 1000 * fTvol * Tr) + fh_3 * (
    ac_3_POCPPh + tc["POCP"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_POCPe + tc["POCP"] / 1000 * fTvol * Te))
BMEF_ac_3_AP = float(
    fs_3 * (ac_3_APs + tc["AP"] / 1000 * fTvol * Ts) + fv_3 * (ac_3_APv + tc["AP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_APr + tc["AP"] / 1000 * fTvol * Tr) + fh_3 * (ac_3_APh + tc["AP"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_APe + tc["AP"] / 1000 * fTvol * Te))
BMEF_ac_3_EP = float(
    fs_3 * (ac_3_EPs + tc["EP"] / 1000 * fTvol * Ts) + fv_3 * (ac_3_EPv + tc["EP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_EPr + tc["EP"] / 1000 * fTvol * Tr) + fh_3 * (ac_3_EPh + tc["EP"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_EPe + tc["EP"] / 1000 * fTvol * Te))
BMEF_ac_3_HTP = float(
    fs_3 * (ac_3_HTPs + tc["HTP"] / 1000 * fTvol * Ts) + fv_3 * (ac_3_HTPv + tc["HTP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_HTPPr + tc["HTP"] / 1000 * fTvol * Tr) + fh_3 * (ac_3_HTPH + tc["HTP"] / 1000 * fTvol * Th) + \

```

```

fe_3 * (ac_3_HTPe + tc["HTP"] / 1000 * fTvol * Te))
BMEF_ac_3_FAETP = float(fs_3 * (ac_3_FAETPs + tc["FAETP"] / 1000 * fTvol * Ts) + fv_3 * (
    ac_3_FAETPv + tc["FAETP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_FAETPr + tc["FAETP"] / 1000 * fTvol * Tr) + fh_3 * (
        ac_3_FAETPh + tc["FAETP"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_FAETPe + tc["FAETP"] / 1000 * fTvol * Te))
BMEF_ac_3_MAETP = float(fs_3 * (ac_3_MAETPs + tc["MAETP"] / 1000 * fTvol * Ts) + fv_3 * (
    ac_3_MAETPv + tc["MAETP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_MAETPr + tc["MAETP"] / 1000 * fTvol * Tr) + fh_3 * (
        ac_3_MAETPh + tc["MAETP"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_MAETPe + tc["MAETP"] / 1000 * fTvol * Te))
BMEF_ac_3_TETP = float(fs_3 * (ac_3_TETPs + tc["TETP"] / 1000 * fTvol * Ts) + fv_3 * (
    ac_3_TETPv + tc["TETP"] / 1000 * fTvol * Tv) + \
    fr_3 * (ac_3_TETPr + tc["TETP"] / 1000 * fTvol * Tr) + fh_3 * (
        ac_3_TETPh + tc["TETP"] / 1000 * fTvol * Th) + \
    fe_3 * (ac_3_TETPe + tc["TETP"] / 1000 * fTvol * Te))

MEFeh_ADPc_3 = float((1 + Ogen) * (BMEF_pc_3_ADPc + tc["ADPc"] / 1000 * Tb * fTvol + BMEF_ac_3_ADPc))
MEFeh_ADPf_3 = float((1 + Ogen) * (BMEF_pc_3_ADPf + tc["ADPf"] / 1000 * Tb * fTvol + BMEF_ac_3_ADPf))
MEFeh_GWP_3 = float((1 + Ogen) * (BMEF_pc_3_GWP + tc["GWP"] / 1000 * Tb * fTvol + BMEF_ac_3_GWP))
MEFeh_ODP_3 = float((1 + Ogen) * (BMEF_pc_3_ODP + tc["ODP"] / 1000 * Tb * fTvol + BMEF_ac_3_ODP))
MEFeh_POCP_3 = float((1 + Ogen) * (BMEF_pc_3_POCP + tc["POCP"] / 1000 * Tb * fTvol + BMEF_ac_3_POCP))
MEFeh_AP_3 = float((1 + Ogen) * (BMEF_pc_3_AP + tc["AP"] / 1000 * Tb * fTvol + BMEF_ac_3_AP))
MEFeh_EP_3 = float((1 + Ogen) * (BMEF_pc_3_EP + tc["EP"] / 1000 * Tb * fTvol + BMEF_ac_3_EP))
MEFeh_HTP_3 = float((1 + Ogen) * (BMEF_pc_3_HTP + tc["HTP"] / 1000 * Tb * fTvol + BMEF_ac_3_HTP))
MEFeh_FAETP_3 = float((1 + Ogen) * (BMEF_pc_3_FAETP + tc["FAETP"] / 1000 * Tb * fTvol + BMEF_ac_3_FAETP))
MEFeh_MAETP_3 = float((1 + Ogen) * (BMEF_pc_3_MAETP + tc["MAETP"] / 1000 * Tb * fTvol + BMEF_ac_3_MAETP))
MEFeh_TETP_3 = float((1 + Ogen) * (BMEF_pc_3_TETP + tc["TETP"] / 1000 * Tb * fTvol + BMEF_ac_3_TETP))

MEFo_3_ADPc = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_ADPc_3
MEFo_3_ADPf = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_ADPf_3
MEFo_3_GWP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_GWP_3
MEFo_3_ODP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_ODP_3
MEFo_3_POCP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_POCP_3
MEFo_3_AP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_AP_3
MEFo_3_EP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_EP_3
MEFo_3_HTP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_HTP_3
MEFo_3_FAETP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_FAETP_3
MEFo_3_MAETP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_MAETP_3
MEFo_3_TETP = hvh_3 * Sftot * (1 + Bafv_3) * Ldf * MEFeh_TETP_3

MEFo_ADPc = MEFo_1_ADPc + MEFo_2_ADPc + MEFo_3_ADPc
MEFo_ADPf = MEFo_1_ADPf + MEFo_2_ADPf + MEFo_3_ADPf
MEFo_GWP = MEFo_1_GWP + MEFo_2_GWP + MEFo_3_GWP
MEFo_ODP = MEFo_1_ODP + MEFo_2_ODP + MEFo_3_ODP
MEFo_POCP = MEFo_1_POCP + MEFo_2_POCP + MEFo_3_POCP
MEFo_AP = MEFo_1_AP + MEFo_2_AP + MEFo_3_AP
MEFo_EP = MEFo_1_EP + MEFo_2_EP + MEFo_3_EP
MEFo_HTP = MEFo_1_HTP + MEFo_2_HTP + MEFo_3_HTP
MEFo_FAETP = MEFo_1_FAETP + MEFo_2_FAETP + MEFo_3_FAETP
MEFo_MAETP = MEFo_1_MAETP + MEFo_2_MAETP + MEFo_3_MAETP
MEFo_TETP = MEFo_1_TETP + MEFo_2_TETP + MEFo_3_TETP
if numberofcomponents > 3:
    Type_4 = (Type + "-4")
    pc_4 = components.loc[Type_4, 'pc']
    ac_4 = components.loc[Type_4, 'ac']
    hvh_4 = components.loc[Type_4, 'hvh']
    Bafv_4 = components.loc[Type_4, 'Bafv']
    fs_4 = components.loc[Type_4, 'fs']
    fv_4 = components.loc[Type_4, 'fv']
    fr_4 = components.loc[Type_4, 'fr']
    fh_4 = components.loc[Type_4, 'fh']
    fe_4 = components.loc[Type_4, 'fe']
    BUPC_4 = components.loc[Type_4, "base_unit_pc_component"]
    BUAC_4 = components.loc[Type_4, "base_unit_ac_component"]

if ac_4 == "00" or BUPC_4 != "kg":
    BMEF_ac_4 = 0
else:
    ac_4_ADPcs = ac.loc[ac_4, "ADPcs"]
    ac_4_ADPfs = ac.loc[ac_4, "ADPfs"]

```

```

ac_4_GWPs = ac.loc[ac_4, "GWPs"]
ac_4_ODPs = ac.loc[ac_4, "ODPs"]
ac_4_POCPs = ac.loc[ac_4, "POCPs"]
ac_4_APs = ac.loc[ac_4, "APs"]
ac_4_EPs = ac.loc[ac_4, "EPs"]
ac_4_HTPs = ac.loc[ac_4, "HTPs"]
ac_4_FAETPs = ac.loc[ac_4, "FAETPs"]
ac_4_MAETPs = ac.loc[ac_4, "MAETPs"]
ac_4_TETPs = ac.loc[ac_4, "TETPs"]
ac_4_ADPCv = ac.loc[ac_4, "ADPCv"]
ac_4_ADPFv = ac.loc[ac_4, "ADPFv"]
ac_4_GWpv = ac.loc[ac_4, "GWpv"]
ac_4_ODPv = ac.loc[ac_4, "ODPv"]
ac_4_POCPv = ac.loc[ac_4, "POCPv"]
ac_4_APv = ac.loc[ac_4, "APv"]
ac_4_EPv = ac.loc[ac_4, "EPv"]
ac_4_HTPv = ac.loc[ac_4, "HTPv"]
ac_4_FAETPv = ac.loc[ac_4, "FAETPv"]
ac_4_MAETPv = ac.loc[ac_4, "MAETPv"]
ac_4_TETPv = ac.loc[ac_4, "TETPv"]
ac_4_ADPCr = ac.loc[ac_4, "ADPCr"]
ac_4_ADPFr = ac.loc[ac_4, "ADPFr"]
ac_4_GWPr = ac.loc[ac_4, "GWPr"]
ac_4_ODPr = ac.loc[ac_4, "ODPr"]
ac_4_POCPr = ac.loc[ac_4, "POCPr"]
ac_4_APr = ac.loc[ac_4, "APr"]
ac_4_EPr = ac.loc[ac_4, "EPr"]
ac_4_HTPr = ac.loc[ac_4, "HTPr"]
ac_4_FAETPr = ac.loc[ac_4, "FAETPr"]
ac_4_MAETPr = ac.loc[ac_4, "MAETPr"]
ac_4_TETPr = ac.loc[ac_4, "TETPr"]
ac_4_ADPh = ac.loc[ac_4, "ADPh"]
ac_4_ADPFh = ac.loc[ac_4, "ADPFh"]
ac_4_GWPh = ac.loc[ac_4, "GWPh"]
ac_4_ODPh = ac.loc[ac_4, "ODPh"]
ac_4_POCPH = ac.loc[ac_4, "POCPH"]
ac_4_APh = ac.loc[ac_4, "APh"]
ac_4_EPh = ac.loc[ac_4, "EPh"]
ac_4_HTPH = ac.loc[ac_4, "HTPH"]
ac_4_FAETPh = ac.loc[ac_4, "FAETPh"]
ac_4_MAETPh = ac.loc[ac_4, "MAETPh"]
ac_4_TETPh = ac.loc[ac_4, "TETPh"]
ac_4_ADPCe = ac.loc[ac_4, "ADPCe"]
ac_4_ADPFe = ac.loc[ac_4, "ADPFe"]
ac_4_GWPe = ac.loc[ac_4, "GWPe"]
ac_4_ODPe = ac.loc[ac_4, "ODPe"]
ac_4_POCPe = ac.loc[ac_4, "POCPe"]
ac_4_APe = ac.loc[ac_4, "APe"]
ac_4_EPe = ac.loc[ac_4, "EPe"]
ac_4_HTPe = ac.loc[ac_4, "HTPe"]
ac_4_FAETPe = ac.loc[ac_4, "FAETPe"]
ac_4_MAETPe = ac.loc[ac_4, "MAETPe"]
ac_4_TETPe = ac.loc[ac_4, "TETPe"]
BMEF_pc_4_ADPC = pc.loc[pc_4, "ADPC"]
BMEF_pc_4_ADPF = pc.loc[pc_4, "ADPF"]
BMEF_pc_4_GWP = pc.loc[pc_4, "GWP"]
BMEF_pc_4_ODP = pc.loc[pc_4, "ODP"]
BMEF_pc_4_POCP = pc.loc[pc_4, "POCP"]
BMEF_pc_4_AP = pc.loc[pc_4, "AP"]
BMEF_pc_4_EP = pc.loc[pc_4, "EP"]
BMEF_pc_4_HTP = pc.loc[pc_4, "HTP"]
BMEF_pc_4_FAETP = pc.loc[pc_4, "FAETP"]
BMEF_pc_4_MAETP = pc.loc[pc_4, "MAETP"]
BMEF_pc_4_TETP = pc.loc[pc_4, "TETP"]

BMEF_ac_4_ADPC = float(fs_4 * (ac_4_ADPCs + tc["ADPC"] / 1000 * fTvol * Ts) + fv_4 * (
    ac_4_ADPCv + tc["ADPC"] / 1000 * fTvol * Tv) + \
    fr_4 * (ac_4_ADPCr + tc["ADPC"] / 1000 * fTvol * Tr) + fh_4 * (
    ac_4_ADPh + tc["ADPC"] / 1000 * fTvol * Th) + \
    fe_4 * (ac_4_ADPCe + tc["ADPC"] / 1000 * fTvol * Te))
BMEF_ac_4_ADPF = float(fs_4 * (ac_4_ADPFs + tc["ADPF"] / 1000 * fTvol * Ts) + fv_4 * (

```

```

ac_4_ADPPfv + tc["ADPPf"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_ADPPfr + tc["ADPPf"] / 1000 * fTvol * Tr) + fh_4 * (
ac_4_ADPPfh + tc["ADPPf"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_ADPPfe + tc["ADPPf"] / 1000 * fTvol * Te))
BMEF_ac_4_GWP = float(
fs_4 * (ac_4_GWPs + tc["GWP"] / 1000 * fTvol * Ts) + fv_4 * (ac_4_GWPs + tc["GWP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_GWPr + tc["GWP"] / 1000 * fTvol * Tr) + fh_4 * (ac_4_GWPh + tc["GWP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_GWPe + tc["GWP"] / 1000 * fTvol * Te))
BMEF_ac_4_ODP = float(
fs_4 * (ac_4_ODPs + tc["ODP"] / 1000 * fTvol * Ts) + fv_4 * (ac_4_ODPs + tc["ODP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_ODPr + tc["ODP"] / 1000 * fTvol * Tr) + fh_4 * (ac_4_ODPh + tc["ODP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_ODPe + tc["ODP"] / 1000 * fTvol * Te))
BMEF_ac_4_POCP = float(fs_4 * (ac_4_POCPs + tc["POCP"] / 1000 * fTvol * Ts) + fv_4 * (
ac_4_POCPv + tc["POCP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_POCPPr + tc["POCP"] / 1000 * fTvol * Tr) + fh_4 * (
ac_4_POCPPh + tc["POCP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_POCPPe + tc["POCP"] / 1000 * fTvol * Te))
BMEF_ac_4_AP = float(
fs_4 * (ac_4_APs + tc["AP"] / 1000 * fTvol * Ts) + fv_4 * (ac_4_APv + tc["AP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_APr + tc["AP"] / 1000 * fTvol * Tr) + fh_4 * (ac_4_APh + tc["AP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_APe + tc["AP"] / 1000 * fTvol * Te))
BMEF_ac_4_EP = float(
fs_4 * (ac_4_EPs + tc["EP"] / 1000 * fTvol * Ts) + fv_4 * (ac_4_EPs + tc["EP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_EPr + tc["EP"] / 1000 * fTvol * Tr) + fh_4 * (ac_4_EPh + tc["EP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_EPe + tc["EP"] / 1000 * fTvol * Te))
BMEF_ac_4_HTP = float(
fs_4 * (ac_4_HTPs + tc["HTP"] / 1000 * fTvol * Ts) + fv_4 * (ac_4_HTPv + tc["HTP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_HTPPr + tc["HTP"] / 1000 * fTvol * Tr) + fh_4 * (ac_4_HTPPh + tc["HTP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_HTPPe + tc["HTP"] / 1000 * fTvol * Te))
BMEF_ac_4_FAETP = float(fs_4 * (ac_4_FAETPs + tc["FAETP"] / 1000 * fTvol * Ts) + fv_4 * (
ac_4_FAETPv + tc["FAETP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_FAETPr + tc["FAETP"] / 1000 * fTvol * Tr) + fh_4 * (
ac_4_FAETPh + tc["FAETP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_FAETPe + tc["FAETP"] / 1000 * fTvol * Te))
BMEF_ac_4_MAETP = float(fs_4 * (ac_4_MAETPs + tc["MAETP"] / 1000 * fTvol * Ts) + fv_4 * (
ac_4_MAETPv + tc["MAETP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_MAETPr + tc["MAETP"] / 1000 * fTvol * Tr) + fh_4 * (
ac_4_MAETPh + tc["MAETP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_MAETPe + tc["MAETP"] / 1000 * fTvol * Te))
BMEF_ac_4_TETP = float(fs_4 * (ac_4_TETPs + tc["TETP"] / 1000 * fTvol * Ts) + fv_4 * (
ac_4_TETPv + tc["TETP"] / 1000 * fTvol * Tv) + \
fr_4 * (ac_4_TETPr + tc["TETP"] / 1000 * fTvol * Tr) + fh_4 * (
ac_4_TETPh + tc["TETP"] / 1000 * fTvol * Th) + \
fe_4 * (ac_4_TETPe + tc["TETP"] / 1000 * fTvol * Te))

MEFeh_ADPPc_4 = float((1 + Ogen) * (BMEF_pc_4_ADPPc + tc["ADPPc"] / 1000 * Tb * fTvol + BMEF_ac_4_ADPPc))
MEFeh_ADPPf_4 = float((1 + Ogen) * (BMEF_pc_4_ADPPf + tc["ADPPf"] / 1000 * Tb * fTvol + BMEF_ac_4_ADPPf))
MEFeh_GWP_4 = float((1 + Ogen) * (BMEF_pc_4_GWP + tc["GWP"] / 1000 * Tb * fTvol + BMEF_ac_4_GWP))
MEFeh_ODP_4 = float((1 + Ogen) * (BMEF_pc_4_ODP + tc["ODP"] / 1000 * Tb * fTvol + BMEF_ac_4_ODP))
MEFeh_POCP_4 = float((1 + Ogen) * (BMEF_pc_4_POCP + tc["POCP"] / 1000 * Tb * fTvol + BMEF_ac_4_POCP))
MEFeh_AP_4 = float((1 + Ogen) * (BMEF_pc_4_AP + tc["AP"] / 1000 * Tb * fTvol + BMEF_ac_4_AP))
MEFeh_EP_4 = float((1 + Ogen) * (BMEF_pc_4_EP + tc["EP"] / 1000 * Tb * fTvol + BMEF_ac_4_EP))
MEFeh_HTP_4 = float((1 + Ogen) * (BMEF_pc_4_HTP + tc["HTP"] / 1000 * Tb * fTvol + BMEF_ac_4_HTP))
MEFeh_FAETP_4 = float((1 + Ogen) * (BMEF_pc_4_FAETP + tc["FAETP"] / 1000 * Tb * fTvol + BMEF_ac_4_FAETP))
MEFeh_MAETP_4 = float((1 + Ogen) * (BMEF_pc_4_MAETP + tc["MAETP"] / 1000 * Tb * fTvol + BMEF_ac_4_MAETP))
MEFeh_TETP_4 = float((1 + Ogen) * (BMEF_pc_4_TETP + tc["TETP"] / 1000 * Tb * fTvol + BMEF_ac_4_TETP))

MEFo_4_ADPPc = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_ADPPc_4
MEFo_4_ADPPf = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_ADPPf_4
MEFo_4_GWP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_GWP_4
MEFo_4_ODP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_ODP_4
MEFo_4_POCP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_POCP_4
MEFo_4_AP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_AP_4
MEFo_4_EP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_EP_4
MEFo_4_HTP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_HTP_4
MEFo_4_FAETP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_FAETP_4
MEFo_4_MAETP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_MAETP_4
MEFo_4_TETP = hvh_4 * Sftot * (1 + Bafv_4) * Ldf * MEFeh_TETP_4

MEFo_ADPPc = MEFo_1_ADPPc + MEFo_2_ADPPc + MEFo_3_ADPPc + MEFo_4_ADPPc
MEFo_ADPPf = MEFo_1_ADPPf + MEFo_2_ADPPf + MEFo_3_ADPPf + MEFo_4_ADPPf

```



```

MEFo_GWP = MEFo_1_GWP + MEFo_2_GWP + MEFo_3_GWP + MEFo_4_GWP
MEFo_ODP = MEFo_1_ODP + MEFo_2_ODP + MEFo_3_ODP + MEFo_4_ODP
MEFo_POCP = MEFo_1_POCP + MEFo_2_POCP + MEFo_3_POCP + MEFo_4_POCP
MEFo_AP = MEFo_1_AP + MEFo_2_AP + MEFo_3_AP + MEFo_4_AP
MEFo_EP = MEFo_1_EP + MEFo_2_EP + MEFo_3_EP + MEFo_4_EP
MEFo_HTP = MEFo_1_HTP + MEFo_2_HTP + MEFo_3_HTP + MEFo_4_HTP
MEFo_FAETP = MEFo_1_FAETP + MEFo_2_FAETP + MEFo_3_FAETP + MEFo_4_FAETP
MEFo_MAETP = MEFo_1_MAETP + MEFo_2_MAETP + MEFo_3_MAETP + MEFo_4_MAETP
MEFo_TETP = MEFo_1_TETP + MEFo_2_TETP + MEFo_3_TETP + MEFo_4_TETP

```

```

if numberofcomponents > 4:

```

```

    Type_5 = (Type + "-5")
    pc_5 = components.loc[Type_5, 'pc']
    ac_5 = components.loc[Type_5, 'ac']
    hvh_5 = components.loc[Type_5, 'hvh']
    Bafv_5 = components.loc[Type_5, 'Bafv']
    fs_5 = components.loc[Type_5, 'fs']
    fv_5 = components.loc[Type_5, 'fv']
    fr_5 = components.loc[Type_5, 'fr']
    fh_5 = components.loc[Type_5, 'fh']
    fe_5 = components.loc[Type_5, 'fe']
    BUPC_5 = components.loc[Type_5, "base_unit_pc_component"]
    BUAC_5 = components.loc[Type_5, "base_unit_ac_component"]

```

```

if ac_5 == "00" or BUPC_5 != "kg":

```

```

    BMEF_ac_5 = 0
else:
    ac_5_ADPCs = ac.loc[ac_5, "ADPCs"]
    ac_5_ADPFs = ac.loc[ac_5, "ADPFs"]
    ac_5_GWPs = ac.loc[ac_5, "GWPs"]
    ac_5_ODPs = ac.loc[ac_5, "ODPs"]
    ac_5_POCPs = ac.loc[ac_5, "POCPs"]
    ac_5_APs = ac.loc[ac_5, "APs"]
    ac_5_EPs = ac.loc[ac_5, "EPs"]
    ac_5_HTPs = ac.loc[ac_5, "HTPs"]
    ac_5_FAETPs = ac.loc[ac_5, "FAETPs"]
    ac_5_MAETPs = ac.loc[ac_5, "MAETPs"]
    ac_5_TETPs = ac.loc[ac_5, "TETPs"]
    ac_5_ADPCv = ac.loc[ac_5, "ADPCv"]
    ac_5_ADPFv = ac.loc[ac_5, "ADPFv"]
    ac_5_GWPv = ac.loc[ac_5, "GWPv"]
    ac_5_ODPv = ac.loc[ac_5, "ODPv"]
    ac_5_POCPv = ac.loc[ac_5, "POCPv"]
    ac_5_APv = ac.loc[ac_5, "APv"]
    ac_5_EPv = ac.loc[ac_5, "EPv"]
    ac_5_HTPv = ac.loc[ac_5, "HTPv"]
    ac_5_FAETPv = ac.loc[ac_5, "FAETPv"]
    ac_5_MAETPv = ac.loc[ac_5, "MAETPv"]
    ac_5_TETPv = ac.loc[ac_5, "TETPv"]
    ac_5_ADPCr = ac.loc[ac_5, "ADPCr"]
    ac_5_ADPfr = ac.loc[ac_5, "ADPfr"]
    ac_5_GWPr = ac.loc[ac_5, "GWPr"]
    ac_5_ODPr = ac.loc[ac_5, "ODPr"]
    ac_5_POCPv = ac.loc[ac_5, "POCPv"]
    ac_5_APv = ac.loc[ac_5, "APv"]
    ac_5_EPv = ac.loc[ac_5, "EPv"]
    ac_5_HTPv = ac.loc[ac_5, "HTPv"]
    ac_5_FAETPr = ac.loc[ac_5, "FAETPr"]
    ac_5_MAETPr = ac.loc[ac_5, "MAETPr"]
    ac_5_TETPr = ac.loc[ac_5, "TETPr"]
    ac_5_ADPCr = ac.loc[ac_5, "ADPCr"]
    ac_5_ADPfr = ac.loc[ac_5, "ADPfr"]
    ac_5_GWPr = ac.loc[ac_5, "GWPr"]
    ac_5_ODPr = ac.loc[ac_5, "ODPr"]
    ac_5_POCPv = ac.loc[ac_5, "POCPv"]
    ac_5_APv = ac.loc[ac_5, "APv"]
    ac_5_EPv = ac.loc[ac_5, "EPv"]
    ac_5_HTPv = ac.loc[ac_5, "HTPv"]
    ac_5_FAETPh = ac.loc[ac_5, "FAETPh"]
    ac_5_MAETPh = ac.loc[ac_5, "MAETPh"]
    ac_5_TETPh = ac.loc[ac_5, "TETPh"]

```

```

ac_5_ADPce = ac.loc[ac_5, "ADPce"]
ac_5_ADPfe = ac.loc[ac_5, "ADPfe"]
ac_5_GWPe = ac.loc[ac_5, "GWPe"]
ac_5_ODPe = ac.loc[ac_5, "ODPe"]
ac_5_POCPe = ac.loc[ac_5, "POCPe"]
ac_5_APe = ac.loc[ac_5, "APe"]
ac_5_EPe = ac.loc[ac_5, "EPe"]
ac_5_HTPe = ac.loc[ac_5, "HTPe"]
ac_5_FAETPe = ac.loc[ac_5, "FAETPe"]
ac_5_MAETPe = ac.loc[ac_5, "MAETPe"]
ac_5_TETPe = ac.loc[ac_5, "TETPe"]
BMEF_pc_5_ADPc = pc.loc[pc_5, "ADPc"]
BMEF_pc_5_ADPf = pc.loc[pc_5, "ADPf"]
BMEF_pc_5_GWP = pc.loc[pc_5, "GWP"]
BMEF_pc_5_ODP = pc.loc[pc_5, "ODP"]
BMEF_pc_5_POCP = pc.loc[pc_5, "POCP"]
BMEF_pc_5_AP = pc.loc[pc_5, "AP"]
BMEF_pc_5_EP = pc.loc[pc_5, "EP"]
BMEF_pc_5_HTP = pc.loc[pc_5, "HTP"]
BMEF_pc_5_FAETP = pc.loc[pc_5, "FAETP"]
BMEF_pc_5_MAETP = pc.loc[pc_5, "MAETP"]
BMEF_pc_5_TETP = pc.loc[pc_5, "TETP"]

BMEF_ac_5_ADPc = float(fs_5 * (ac_5_ADPcs + tc["ADPc"] / 1000 * fTvol * Ts) + fv_5 * (
    ac_5_ADPcv + tc["ADPc"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_ADPcr + tc["ADPc"] / 1000 * fTvol * Tr) + fh_5 * (
    ac_5_ADPch + tc["ADPc"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_ADPce + tc["ADPc"] / 1000 * fTvol * Te))
BMEF_ac_5_ADPf = float(fs_5 * (ac_5_ADPfs + tc["ADPf"] / 1000 * fTvol * Ts) + fv_5 * (
    ac_5_ADPfv + tc["ADPf"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_ADPfr + tc["ADPf"] / 1000 * fTvol * Tr) + fh_5 * (
    ac_5_ADPfh + tc["ADPf"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_ADPfe + tc["ADPf"] / 1000 * fTvol * Te))
BMEF_ac_5_GWP = float(
    fs_5 * (ac_5_GWPs + tc["GWP"] / 1000 * fTvol * Ts) + fv_5 * (ac_5_GWpv + tc["GWP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_GWPr + tc["GWP"] / 1000 * fTvol * Tr) + fh_5 * (ac_5_GWPh + tc["GWP"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_GWPe + tc["GWP"] / 1000 * fTvol * Te))
BMEF_ac_5_ODP = float(
    fs_5 * (ac_5_ODPs + tc["ODP"] / 1000 * fTvol * Ts) + fv_5 * (ac_5_ODpv + tc["ODP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_ODPr + tc["ODP"] / 1000 * fTvol * Tr) + fh_5 * (ac_5_ODPh + tc["ODP"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_ODPe + tc["ODP"] / 1000 * fTvol * Te))
BMEF_ac_5_POCP = float(fs_5 * (ac_5_POCPs + tc["POCP"] / 1000 * fTvol * Ts) + fv_5 * (
    ac_5_POCPv + tc["POCP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_POCPPr + tc["POCP"] / 1000 * fTvol * Tr) + fh_5 * (
    ac_5_POCPPh + tc["POCP"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_POCPe + tc["POCP"] / 1000 * fTvol * Te))
BMEF_ac_5_AP = float(
    fs_5 * (ac_5_APs + tc["AP"] / 1000 * fTvol * Ts) + fv_5 * (ac_5_APv + tc["AP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_APPr + tc["AP"] / 1000 * fTvol * Tr) + fh_5 * (ac_5_APPh + tc["AP"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_APe + tc["AP"] / 1000 * fTvol * Te))
BMEF_ac_5_EP = float(
    fs_5 * (ac_5_EPs + tc["EP"] / 1000 * fTvol * Ts) + fv_5 * (ac_5_EPv + tc["EP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_EPr + tc["EP"] / 1000 * fTvol * Tr) + fh_5 * (ac_5_EPh + tc["EP"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_EPe + tc["EP"] / 1000 * fTvol * Te))
BMEF_ac_5_HTP = float(
    fs_5 * (ac_5_HTPs + tc["HTP"] / 1000 * fTvol * Ts) + fv_5 * (ac_5_HTPv + tc["HTP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_HTPPr + tc["HTP"] / 1000 * fTvol * Tr) + fh_5 * (ac_5_HTPPh + tc["HTP"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_HTPe + tc["HTP"] / 1000 * fTvol * Te))
BMEF_ac_5_FAETP = float(fs_5 * (ac_5_FAETPs + tc["FAETP"] / 1000 * fTvol * Ts) + fv_5 * (
    ac_5_FAETPv + tc["FAETP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_FAETPr + tc["FAETP"] / 1000 * fTvol * Tr) + fh_5 * (
    ac_5_FAETPh + tc["FAETP"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_FAETPe + tc["FAETP"] / 1000 * fTvol * Te))
BMEF_ac_5_MAETP = float(fs_5 * (ac_5_MAETPs + tc["MAETP"] / 1000 * fTvol * Ts) + fv_5 * (
    ac_5_MAETPv + tc["MAETP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_MAETPr + tc["MAETP"] / 1000 * fTvol * Tr) + fh_5 * (
    ac_5_MAETPh + tc["MAETP"] / 1000 * fTvol * Th) + \
    fe_5 * (ac_5_MAETPe + tc["MAETP"] / 1000 * fTvol * Te))
BMEF_ac_5_TETP = float(fs_5 * (ac_5_TETPs + tc["TETP"] / 1000 * fTvol * Ts) + fv_5 * (
    ac_5_TETPv + tc["TETP"] / 1000 * fTvol * Tv) + \
    fr_5 * (ac_5_TETPr + tc["TETP"] / 1000 * fTvol * Tr) + fh_5 * (

```

```

ac_5_TETPh + tc["TETP"] / 1000 * fTvol * Th) + \
fe_5 * (ac_5_TETPe + tc["TETP"] / 1000 * fTvol * Te))

MEFeh_ADPC_5 = float((1 + Ogen) * (BMEF_pc_5_ADPC + tc["ADPC"] / 1000 * Tb * fTvol + BMEF_ac_5_ADPC))
MEFeh_ADPF_5 = float((1 + Ogen) * (BMEF_pc_5_ADPF + tc["ADPF"] / 1000 * Tb * fTvol + BMEF_ac_5_ADPF))
MEFeh_GWP_5 = float((1 + Ogen) * (BMEF_pc_5_GWP + tc["GWP"] / 1000 * Tb * fTvol + BMEF_ac_5_GWP))
MEFeh_ODP_5 = float((1 + Ogen) * (BMEF_pc_5_ODP + tc["ODP"] / 1000 * Tb * fTvol + BMEF_ac_5_ODP))
MEFeh_POCP_5 = float((1 + Ogen) * (BMEF_pc_5_POCP + tc["POCP"] / 1000 * Tb * fTvol + BMEF_ac_5_POCP))
MEFeh_AP_5 = float((1 + Ogen) * (BMEF_pc_5_AP + tc["AP"] / 1000 * Tb * fTvol + BMEF_ac_5_AP))
MEFeh_EP_5 = float((1 + Ogen) * (BMEF_pc_5_EP + tc["EP"] / 1000 * Tb * fTvol + BMEF_ac_5_EP))
MEFeh_HTP_5 = float((1 + Ogen) * (BMEF_pc_5_HTP + tc["HTP"] / 1000 * Tb * fTvol + BMEF_ac_5_HTP))
MEFeh_FAETP_5 = float((1 + Ogen) * (BMEF_pc_5_FAETP + tc["FAETP"] / 1000 * Tb * fTvol + BMEF_ac_5_FAETP))
MEFeh_MAETP_5 = float((1 + Ogen) * (BMEF_pc_5_MAETP + tc["MAETP"] / 1000 * Tb * fTvol + BMEF_ac_5_MAETP))
MEFeh_TETP_5 = float((1 + Ogen) * (BMEF_pc_5_TETP + tc["TETP"] / 1000 * Tb * fTvol + BMEF_ac_5_TETP))

MEFo_5_ADPC = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_ADPC_5
MEFo_5_ADPF = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_ADPF_5
MEFo_5_GWP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_GWP_5
MEFo_5_ODP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_ODP_5
MEFo_5_POCP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_POCP_5
MEFo_5_AP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_AP_5
MEFo_5_EP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_EP_5
MEFo_5_HTP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_HTP_5
MEFo_5_FAETP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_FAETP_5
MEFo_5_MAETP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_MAETP_5
MEFo_5_TETP = hvh_5 * Sftot * (1 + Bafv_5) * Ldf * MEFeh_TETP_5

MEFo_ADPC = MEFo_1_ADPC + MEFo_2_ADPC + MEFo_3_ADPC + MEFo_4_ADPC + MEFo_5_ADPC
MEFo_ADPF = MEFo_1_ADPF + MEFo_2_ADPF + MEFo_3_ADPF + MEFo_4_ADPF + MEFo_5_ADPF
MEFo_GWP = MEFo_1_GWP + MEFo_2_GWP + MEFo_3_GWP + MEFo_4_GWP + MEFo_5_GWP
MEFo_ODP = MEFo_1_ODP + MEFo_2_ODP + MEFo_3_ODP + MEFo_4_ODP + MEFo_5_ODP
MEFo_POCP = MEFo_1_POCP + MEFo_2_POCP + MEFo_3_POCP + MEFo_4_POCP + MEFo_5_POCP
MEFo_AP = MEFo_1_AP + MEFo_2_AP + MEFo_3_AP + MEFo_4_AP + MEFo_5_AP
MEFo_EP = MEFo_1_EP + MEFo_2_EP + MEFo_3_EP + MEFo_4_EP + MEFo_5_EP
MEFo_HTP = MEFo_1_HTP + MEFo_2_HTP + MEFo_3_HTP + MEFo_4_HTP + MEFo_5_HTP
MEFo_FAETP = MEFo_1_FAETP + MEFo_2_FAETP + MEFo_3_FAETP + MEFo_4_FAETP + MEFo_5_FAETP
MEFo_MAETP = MEFo_1_MAETP + MEFo_2_MAETP + MEFo_3_MAETP + MEFo_4_MAETP + MEFo_5_MAETP
MEFo_TETP = MEFo_1_TETP + MEFo_2_TETP + MEFo_3_TETP + MEFo_4_TETP + MEFo_5_TETP

MEFp_GWP = MEFo_GWP
MEFp_ODP = MEFo_ODP
MEFp_POCP = MEFo_POCP
MEFp_AP = MEFo_AP
MEFp_EP = MEFo_EP
MEFp_HTP = MEFo_HTP
MEFp_FAETP = MEFo_FAETP
MEFp_MAETP = MEFo_MAETP
MEFp_TETP = MEFo_TETP
MEFp_ADPC = MEFo_ADPC
MEFp_ADPF = MEFo_ADPF

##### SPECIFIC #####
MEFp_s_GWP = max(1, float(LDg/LDp)) * MEFp_GWP * surface_area_spe
MEFp_s_ODP = max(1, float(LDg/LDp)) * MEFp_ODP * surface_area_spe
MEFp_s_POCP = max(1, float(LDg/LDp)) * MEFp_POCP * surface_area_spe
MEFp_s_AP = max(1, float(LDg/LDp)) * MEFp_AP * surface_area_spe
MEFp_s_EP = max(1, float(LDg/LDp)) * MEFp_EP * surface_area_spe
MEFp_s_HTP = max(1, float(LDg/LDp)) * MEFp_HTP * surface_area_spe
MEFp_s_FAETP = max(1, float(LDg/LDp)) * MEFp_FAETP * surface_area_spe
MEFp_s_MAETP = max(1, float(LDg/LDp)) * MEFp_MAETP * surface_area_spe
MEFp_s_TETP = max(1, float(LDg/LDp)) * MEFp_TETP * surface_area_spe
MEFp_s_ADPC = max(1, float(LDg/LDp)) * MEFp_ADPC * surface_area_spe
MEFp_s_ADPF = max(1, float(LDg/LDp)) * MEFp_ADPF * surface_area_spe

SCp_s_GWP = MEFp_s_GWP * 0.05
SCp_s_ODP = MEFp_s_ODP * 30
SCp_s_POCP = MEFp_s_POCP * 2
SCp_s_AP = MEFp_s_AP * 4
SCp_s_EP = MEFp_s_EP * 9
SCp_s_HTP = MEFp_s_HTP * 0.09
SCp_s_FAETP = MEFp_s_FAETP * 0.03

```

```

SCp_s_MAETP = MEFp_s_MAETP * 0.0001
SCp_s_TETP = MEFp_s_TETP * 0.06
SCp_s_ADPC = MEFp_s_ADPC * 0.16
SCp_s_ADPf = MEFp_s_ADPf * 0.16

SCp_EI_s = SCp_s_GWP + SCp_s_ODP + SCp_s_POCP + SCp_s_AP + SCp_s_EP + SCp_s_HTP + SCp_s_FAETP + SCp_s_MAETP + SCp_s_TETP +
SCp_s_ADPC + SCp_s_ADPf

##### GENERIC #####
MEFp_g_GWP = max(1, float(LDg/LDp)) * MEFp_GWP * surface_area_gen
MEFp_g_ODP = max(1, float(LDg/LDp)) * MEFp_ODP * surface_area_gen
MEFp_g_POCP = max(1, float(LDg/LDp)) * MEFp_POCP * surface_area_gen
MEFp_g_AP = max(1, float(LDg/LDp)) * MEFp_AP * surface_area_gen
MEFp_g_EP = max(1, float(LDg/LDp)) * MEFp_EP * surface_area_gen
MEFp_g_HTP = max(1, float(LDg/LDp)) * MEFp_HTP * surface_area_gen
MEFp_g_FAETP = max(1, float(LDg/LDp)) * MEFp_FAETP * surface_area_gen
MEFp_g_MAETP = max(1, float(LDg/LDp)) * MEFp_MAETP * surface_area_gen
MEFp_g_TETP = max(1, float(LDg/LDp)) * MEFp_TETP * surface_area_gen
MEFp_g_ADPC = max(1, float(LDg/LDp)) * MEFp_ADPC * surface_area_gen
MEFp_g_ADPf = max(1, float(LDg/LDp)) * MEFp_ADPf * surface_area_gen

SCp_g_GWP = MEFp_g_GWP * 0.05
SCp_g_ODP = MEFp_g_ODP * 30
SCp_g_POCP = MEFp_g_POCP * 2
SCp_g_AP = MEFp_g_AP * 4
SCp_g_EP = MEFp_g_EP * 9
SCp_g_HTP = MEFp_g_HTP * 0.09
SCp_g_FAETP = MEFp_g_FAETP * 0.03
SCp_g_MAETP = MEFp_g_MAETP * 0.0001
SCp_g_TETP = MEFp_g_TETP * 0.06
SCp_g_ADPC = MEFp_g_ADPC * 0.16
SCp_g_ADPf = MEFp_g_ADPf * 0.16

SCp_EI_g = SCp_g_GWP + SCp_g_ODP + SCp_g_POCP + SCp_g_AP + SCp_g_EP + SCp_g_HTP + SCp_g_FAETP + SCp_g_MAETP + SCp_g_TETP
+ SCp_g_ADPC + SCp_g_ADPf

##### RESULTS #####
dictionary_SCp_s = [SCp_s_GWP, SCp_s_ODP, SCp_s_POCP, SCp_s_AP, SCp_s_EP, SCp_s_HTP, SCp_s_FAETP, SCp_s_MAETP, SCp_s_TETP,
SCp_s_ADPC, SCp_s_ADPf]
dictionary_MEFp_s = [MEFp_s_GWP, MEFp_s_ODP, MEFp_s_POCP, MEFp_s_AP, MEFp_s_EP, MEFp_s_HTP, MEFp_s_FAETP,
MEFp_s_MAETP, MEFp_s_TETP, MEFp_s_ADPC, MEFp_s_ADPf]
dictionary_SCp_g = [SCp_g_GWP, SCp_g_ODP, SCp_g_POCP, SCp_g_AP, SCp_g_EP, SCp_g_HTP, SCp_g_FAETP, SCp_g_MAETP, SCp_g_TETP,
SCp_g_ADPC, SCp_g_ADPf]
dictionary_MEFp_g = [MEFp_g_GWP, MEFp_g_ODP, MEFp_g_POCP, MEFp_g_AP, MEFp_g_EP, MEFp_g_HTP, MEFp_g_FAETP,
MEFp_g_MAETP, MEFp_g_TETP, MEFp_g_ADPC, MEFp_g_ADPf]
dictionary_MEFp_p = [MEFp_GWP, MEFp_ODP, MEFp_POCP, MEFp_AP, MEFp_EP, MEFp_HTP, MEFp_FAETP, MEFp_MAETP, MEFp_TETP,
MEFp_ADPC, MEFp_ADPf]
Cost_tot_s = Cost * surface_area_spe
Cost_tot_g = Cost * surface_area_gen
SCp_EI = SCp_EI_g + SCp_EI_s
Cost_tot = Cost_tot_g + Cost_tot_s

```

IMPORTCOMPONENTS.PY

```

import importlib
dictionary_empty_start = [0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]
import basevalues
import numpy as np
import sys
sys.path.append('components/')
import EC_2203001
import EC_2203003
import EC_2203005
import EC_2203016
import EC_2203024
import EC_2301072
import EC_2301003
import EC_2301004
import EC_2301012
import EC_2301015
import EC_2801003
import EC_2801028
import EC_2801029
building_type = basevalues.building_type
calculation_choice = basevalues.calculation_choice
building_system = basevalues.building_system
factoreco = basevalues.factorecoimpact
factorenv = basevalues.factorenvimpact
importlib.reload(basevalues)
importlib.reload(EC_2203001)
importlib.reload(EC_2203003)
importlib.reload(EC_2203005)
importlib.reload(EC_2203016)
importlib.reload(EC_2203024)

##### IMPORT OF ALL THE MATERIALS - NOTE HAS TO BE MADE ON THE DICTIONARIES OF MEFp_g and SCp_g --> THESE EQUAL THE
IMPORTS OF DICTIONARIES MEFp_p and SCp_p
### Wallcode 22###
if building_system == "All":
    dictionary_MEFp_p_InteriorWall = EC_2203001.dictionary_MEFp_p, EC_2203003.dictionary_MEFp_p, EC_2203005.dictionary_MEFp_p,
    EC_2203016.dictionary_MEFp_p, EC_2203024.dictionary_MEFp_p
    dictionary_MEFp_g_InteriorWall = EC_2203001.dictionary_MEFp_p+ EC_2203003.dictionary_MEFp_p+ EC_2203005.dictionary_MEFp_p+
    EC_2203016.dictionary_MEFp_p+ EC_2203024.dictionary_MEFp_p
    dictionary_MEFp_s_InteriorWall = EC_2203001.dictionary_MEFp_s, EC_2203003.dictionary_MEFp_s, EC_2203005.dictionary_MEFp_s,
    EC_2203016.dictionary_MEFp_s, EC_2203024.dictionary_MEFp_s

    dictionary_SCp_p_InteriorWall = EC_2203001.dictionary_SCp_p, EC_2203003.dictionary_SCp_p, EC_2203005.dictionary_SCp_p,
    EC_2203016.dictionary_SCp_p, EC_2203024.dictionary_SCp_p
    dictionary_SCp_g_InteriorWall = EC_2203001.dictionary_SCp_p+ EC_2203003.dictionary_SCp_p+ EC_2203005.dictionary_SCp_p+
    EC_2203016.dictionary_SCp_p+ EC_2203024.dictionary_SCp_p
    dictionary_SCp_s_InteriorWall = EC_2203001.dictionary_SCp_s, EC_2203003.dictionary_SCp_s, EC_2203005.dictionary_SCp_s,
    EC_2203016.dictionary_SCp_s, EC_2203024.dictionary_SCp_s
    dictionary_SCp_tot_g_InteriorWall = sum(EC_2203001.dictionary_SCp_p), sum(EC_2203003.dictionary_SCp_p),
    sum(EC_2203005.dictionary_SCp_p), sum(EC_2203016.dictionary_SCp_p), sum(EC_2203024.dictionary_SCp_p)

    dictionary_Type_list_InteriorWall = [EC_2203001.Type, EC_2203003.Type, EC_2203005.Type, EC_2203016.Type, EC_2203024.Type]
    dictionary_CostWall_p_InteriorWall = list(map(int, (EC_2203001.Cost_tot_p, EC_2203003.Cost_tot_p, EC_2203005.Cost_tot_p,
    EC_2203016.Cost_tot_p, EC_2203024.Cost_tot_p)))
    dictionary_CostWall_g_InteriorWall = EC_2203001.Cost_tot_p+ EC_2203003.Cost_tot_p+ EC_2203005.Cost_tot_p+
    EC_2203016.Cost_tot_p+ EC_2203024.Cost_tot_p
    dictionary_CostWall_s_InteriorWall = EC_2203001.Cost_tot_s, EC_2203003.Cost_tot_s, EC_2203005.Cost_tot_s, EC_2203016.Cost_tot_s,
    EC_2203024.Cost_tot_s
    dictionary_SurfaceWall_g_InteriorWall = EC_2203001.surface_area_gen + EC_2203003.surface_area_gen +
    EC_2203005.surface_area_gen + EC_2203016.surface_area_gen + EC_2203024.surface_area_gen
    dictionary_SurfaceWall_g_InteriorWall_ad = 0
    dictionary_SurfaceWall_s_InteriorWall = EC_2203001.surface_area_spe+ EC_2203003.surface_area_spe + EC_2203005.surface_area_spe
    + EC_2203016.surface_area_spe + EC_2203024.surface_area_spe
    elif building_system == "Bricks and concrete":
        dictionary_MEFp_p_InteriorWall = EC_2203003.dictionary_MEFp_p, EC_2203005.dictionary_MEFp_p, EC_2203016.dictionary_MEFp_p,
        EC_2203024.dictionary_MEFp_p
        dictionary_MEFp_g_InteriorWall = EC_2203003.dictionary_MEFp_p+ EC_2203005.dictionary_MEFp_p+ EC_2203016.dictionary_MEFp_p+
        EC_2203024.dictionary_MEFp_p
        dictionary_MEFp_s_InteriorWall = EC_2203003.dictionary_MEFp_s, EC_2203005.dictionary_MEFp_s, EC_2203016.dictionary_MEFp_s,
        EC_2203024.dictionary_MEFp_s

```

```

dictionary_SCp_p_InteriorWall = EC_2203003.dictionary_SCp_p, EC_2203005.dictionary_SCp_p, EC_2203016.dictionary_SCp_p,
EC_2203024.dictionary_SCp_p
dictionary_SCp_g_InteriorWall = EC_2203003.dictionary_SCp_p + EC_2203005.dictionary_SCp_p + EC_2203016.dictionary_SCp_p +
EC_2203024.dictionary_SCp_p
dictionary_SCp_s_InteriorWall = EC_2203003.dictionary_SCp_s, EC_2203005.dictionary_SCp_s, EC_2203016.dictionary_SCp_s,
EC_2203024.dictionary_SCp_s
dictionary_SCp_tot_g_InteriorWall = sum(EC_2203003.dictionary_SCp_p), sum(EC_2203005.dictionary_SCp_p),
sum(EC_2203016.dictionary_SCp_p), sum(EC_2203024.dictionary_SCp_p)

dictionary_Type_list_InteriorWall = [EC_2203003.Type, EC_2203005.Type, EC_2203016.Type, EC_2203024.Type]
dictionary_CostWall_p_InteriorWall = list(map(int, (EC_2203003.Cost_tot_p, EC_2203005.Cost_tot_p, EC_2203016.Cost_tot_p,
EC_2203024.Cost_tot_p)))
dictionary_CostWall_g_InteriorWall = EC_2203003.Cost_tot_p + EC_2203005.Cost_tot_p + EC_2203016.Cost_tot_p +
EC_2203024.Cost_tot_p
dictionary_CostWall_s_InteriorWall = EC_2203003.Cost_tot_s, EC_2203005.Cost_tot_s, EC_2203016.Cost_tot_s, EC_2203024.Cost_tot_s
dictionary_SurfaceWall_g_InteriorWall = EC_2203003.surface_area_gen + EC_2203005.surface_area_gen +
EC_2203016.surface_area_gen + EC_2203024.surface_area_gen
dictionary_SurfaceWall_g_InteriorWall_ad = EC_2203001.surface_area_gen
dictionary_SurfaceWall_s_InteriorWall = EC_2203003.surface_area_spe + EC_2203005.surface_area_spe + EC_2203016.surface_area_spe
+ EC_2203024.surface_area_spe
elif building_system == "Wood":
dictionary_MEFp_p_InteriorWall = [EC_2203001.dictionary_MEFp_p]
dictionary_MEFp_g_InteriorWall = EC_2203001.dictionary_MEFp_p
dictionary_MEFp_s_InteriorWall = dictionary_empty_start, EC_2203001.dictionary_MEFp_s

dictionary_SCp_p_InteriorWall = [EC_2203001.dictionary_SCp_p]
dictionary_SCp_g_InteriorWall = EC_2203001.dictionary_SCp_p
dictionary_SCp_s_InteriorWall = dictionary_empty_start, EC_2203001.dictionary_SCp_s
dictionary_SCp_tot_g_InteriorWall = sum(EC_2203001.dictionary_SCp_p)

dictionary_Type_list_InteriorWall = [EC_2203001.Type]
dictionary_CostWall_p_InteriorWall = [EC_2203001.Cost_tot_p]
dictionary_CostWall_g_InteriorWall = EC_2203001.Cost_tot_p
dictionary_CostWall_s_InteriorWall = EC_2203001.Cost_tot_s
dictionary_SurfaceWall_g_InteriorWall = EC_2203001.surface_area_gen
dictionary_SurfaceWall_g_InteriorWall_ad = EC_2203003.surface_area_spe + EC_2203005.surface_area_spe +
EC_2203016.surface_area_spe + EC_2203024.surface_area_spe
dictionary_SurfaceWall_s_InteriorWall = EC_2203001.surface_area_spe

countofelements_InteriorWall = len(dictionary_MEFp_p_InteriorWall)

if dictionary_SurfaceWall_g_InteriorWall < 0:
dictionary_SurfaceWall_g_InteriorWall = 0
else:
dictionary_SurfaceWall_g_InteriorWall = dictionary_SurfaceWall_g_InteriorWall / int(countofelements_InteriorWall)

if calculation_choice == "Generic":
WallSurfaceGeneric_InteriorWall = dictionary_SurfaceWall_g_InteriorWall + dictionary_SurfaceWall_g_InteriorWall_ad +
dictionary_SurfaceWall_s_InteriorWall
WallSurfaceSpecific_InteriorWall = 0
CostWallGeneric_InteriorWall = dictionary_CostWall_g_InteriorWall / int(countofelements_InteriorWall) *
WallSurfaceGeneric_InteriorWall
CostWallSpecific_InteriorWall = 0
elif calculation_choice == "Specific":
WallSurfaceGeneric_InteriorWall = dictionary_SurfaceWall_g_InteriorWall + dictionary_SurfaceWall_g_InteriorWall_ad
WallSurfaceSpecific_InteriorWall = dictionary_SurfaceWall_s_InteriorWall
CostWallGeneric_InteriorWall = dictionary_CostWall_g_InteriorWall / int(countofelements_InteriorWall) *
WallSurfaceGeneric_InteriorWall
if countofelements_InteriorWall == 1:
CostWallSpecific_InteriorWall = dictionary_CostWall_s_InteriorWall
elif countofelements_InteriorWall > 1:
CostWallSpecific_InteriorWall = sum(dictionary_CostWall_s_InteriorWall)
else:
WallSurfaceGeneric_InteriorWall = 0
WallSurfaceSpecific_InteriorWall = 0
CostWallGeneric_InteriorWall = 0
CostWallSpecific_InteriorWall = 0

dictionary_MEFp_g_InteriorWall = [WallSurfaceGeneric_InteriorWall * i for i in dictionary_MEFp_g_InteriorWall]
dictionary_SCp_g_InteriorWall = [WallSurfaceGeneric_InteriorWall * i for i in dictionary_SCp_g_InteriorWall]

```

```

dictionary_CostWall_g_InteriorWall = [WallSurfaceGeneric_InteriorWall * i for i in dictionary_CostWall_p_InteriorWall]

def divide_list_interiorwall(l, n):
    for i in range(0, len(l), n):
        yield l[i:i + n]
    n = 11
dictionary_MEFP_g_InteriorWall = list(divide_list_interiorwall(dictionary_MEFP_g_InteriorWall, n))
dictionary_SCp_g_InteriorWall = list(divide_list_interiorwall(dictionary_SCp_g_InteriorWall, n))

bestchoice_InteriorWall_test = WallSurfaceGeneric_InteriorWall

if countofelements_InteriorWall == 1:
    if bestchoice_InteriorWall_test > 0:
        bestchoice_envimp_InteriorWall_type = dictionary_Type_list_InteriorWall[0]
        bestchoice_ecoimp_InteriorWall_type = dictionary_Type_list_InteriorWall[0]
        bestchoice_overall_InteriorWall_type = dictionary_Type_list_InteriorWall[0]
    elif bestchoice_InteriorWall_test > 0:
        dictionary_SCp_total_g_InteriorWall = [WallSurfaceGeneric_InteriorWall * i for i in dictionary_SCp_tot_g_InteriorWall]
        bestchoice_envimp_InteriorWall_np = np.array(dictionary_SCp_total_g_InteriorWall)
        bestchoice_envimp_InteriorWall_nrlist_min = bestchoice_envimp_InteriorWall_np.argmin()
        bestchoice_envimp_InteriorWall_nrlist_max = bestchoice_envimp_InteriorWall_np.argmax()
        bestchoice_envimp_InteriorWall_type = dictionary_Type_list_InteriorWall[bestchoice_envimp_InteriorWall_nrlist_min]
        bestchoice_envimp_InteriorWall = dictionary_SCp_total_g_InteriorWall[bestchoice_envimp_InteriorWall_nrlist_min]
        bestchoice_envimp_InteriorWall_max = dictionary_SCp_total_g_InteriorWall[bestchoice_envimp_InteriorWall_nrlist_max]
        bestchoice_ecoimp_InteriorWall_np = np.array(dictionary_CostWall_g_InteriorWall)
        bestchoice_ecoimp_InteriorWall_nrlist_min = bestchoice_ecoimp_InteriorWall_np.argmin()
        bestchoice_ecoimp_InteriorWall_nrlist_max = bestchoice_ecoimp_InteriorWall_np.argmax()
        bestchoice_ecoimp_InteriorWall_type = dictionary_Type_list_InteriorWall[bestchoice_ecoimp_InteriorWall_nrlist_min]
        bestchoice_ecoimp_InteriorWall = dictionary_CostWall_g_InteriorWall[bestchoice_ecoimp_InteriorWall_nrlist_min]
        bestchoice_ecoimp_InteriorWall_max = dictionary_CostWall_g_InteriorWall[bestchoice_ecoimp_InteriorWall_nrlist_max]
        SCp_g_list_InteriorWall_optimal = [x / bestchoice_envimp_InteriorWall_max * factorenv for x in dictionary_SCp_total_g_InteriorWall]
        Cost_g_list_InteriorWall_optimal = [x / bestchoice_ecoimp_InteriorWall_max * factoreco for x in dictionary_CostWall_g_InteriorWall]
        Overall_list_InteriorWall = {"Cost": Cost_g_list_InteriorWall_optimal, "Environment": SCp_g_list_InteriorWall_optimal}
        Overall_list_InteriorWall_optimal = []
        for value in (zip(*list(Overall_list_InteriorWall.values()))):
            Overall_list_InteriorWall_optimal.append(sum(value))
        linearprogramming_InteriorWall = np.array(Overall_list_InteriorWall_optimal)
        bestchoice_overall_InteriorWall_nrlist = linearprogramming_InteriorWall.argmin()
        bestchoice_overall_InteriorWall_type = dictionary_Type_list_InteriorWall[bestchoice_overall_InteriorWall_nrlist]
    else:
        bestchoice_envimp_InteriorWall_type = "-"
        bestchoice_ecoimp_InteriorWall_type = "-"
        bestchoice_overall_InteriorWall_type = "-"

dictionary_MEFP_p_InteriorWall = [sum(i) for i in zip(*dictionary_MEFP_p_InteriorWall)]
GWP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[0]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
ODP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[1]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
POCP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[2]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
AP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[3]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
EP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[4]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
HTP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[5]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
FAETP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[6]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
MAETP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[7]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
TETP_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[8]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
ADPc_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[9]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
ADPf_MEFP_g_InteriorWall = (dictionary_MEFP_p_InteriorWall[10]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
dictionary_MEFP_g_InteriorWall_averages = [GWP_MEFP_g_InteriorWall, ODP_MEFP_g_InteriorWall, POCP_MEFP_g_InteriorWall,
AP_MEFP_g_InteriorWall, EP_MEFP_g_InteriorWall, HTP_MEFP_g_InteriorWall, FAETP_MEFP_g_InteriorWall,
MAETP_MEFP_g_InteriorWall, TETP_MEFP_g_InteriorWall, ADPc_MEFP_g_InteriorWall, ADPf_MEFP_g_InteriorWall]

dictionary_MEFP_s_InteriorWall = [sum(i) for i in zip(*dictionary_MEFP_s_InteriorWall)]
dictionary_MEFP_InteriorWall = dictionary_MEFP_s_InteriorWall, dictionary_MEFP_g_InteriorWall_averages
dictionary_MEFP_InteriorWall = [sum(i) for i in zip(*dictionary_MEFP_InteriorWall)]
GWP_MEFP_InteriorWall = (dictionary_MEFP_InteriorWall[0])
ODP_MEFP_InteriorWall = (dictionary_MEFP_InteriorWall[1])
POCP_MEFP_InteriorWall = (dictionary_MEFP_InteriorWall[2])
AP_MEFP_InteriorWall = (dictionary_MEFP_InteriorWall[3])
EP_MEFP_InteriorWall = (dictionary_MEFP_InteriorWall[4])
HTP_MEFP_InteriorWall = (dictionary_MEFP_InteriorWall[5])
FAETP_MEFP_InteriorWall = (dictionary_MEFP_InteriorWall[6])
MAETP_MEFP_InteriorWall = (dictionary_MEFP_InteriorWall[7])

```



```

TETP_MEFp_InteriorWall = (dictionary_MEFp_InteriorWall[8])
ADPc_MEFp_InteriorWall = (dictionary_MEFp_InteriorWall[9])
ADPf_MEFp_InteriorWall = (dictionary_MEFp_InteriorWall[10])

dictionary_ScP_p_InteriorWall = [sum(i) for i in zip(*dictionary_ScP_p_InteriorWall)]
GWP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[0]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
ODP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[1]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
POCP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[2]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
AP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[3]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
EP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[4]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
HTP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[5]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
FAETP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[6]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
MAETP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[7]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
TETP_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[8]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
ADPc_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[9]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
ADPf_ScP_g_InteriorWall = (dictionary_ScP_p_InteriorWall[10]) / countofelements_InteriorWall * WallSurfaceGeneric_InteriorWall
dictionary_ScP_g_InteriorWall_average = [GWP_ScP_g_InteriorWall, ODP_ScP_g_InteriorWall, POCP_ScP_g_InteriorWall,
AP_ScP_g_InteriorWall, EP_ScP_g_InteriorWall, HTP_ScP_g_InteriorWall, FAETP_ScP_g_InteriorWall, MAETP_ScP_g_InteriorWall,
TETP_ScP_g_InteriorWall, ADPc_ScP_g_InteriorWall, ADPf_ScP_g_InteriorWall]

dictionary_ScP_g_InteriorWall_totals = (list(sum(i) for i in dictionary_ScP_g_InteriorWall))
dictionary_ScP_g_InteriorWall_totals = [round(x,2) for x in dictionary_ScP_g_InteriorWall_totals]
dictionary_ScP_s_InteriorWall_totals = (list(sum(i) for i in dictionary_ScP_s_InteriorWall))
dictionary_ScP_s_InteriorWall_totals = [round(x,2) for x in dictionary_ScP_s_InteriorWall_totals]

dictionary_ScP_s_InteriorWall = [sum(i) for i in zip(*dictionary_ScP_s_InteriorWall)]
dictionary_ScP_InteriorWall = dictionary_ScP_s_InteriorWall, dictionary_ScP_g_InteriorWall_average
dictionary_ScP_InteriorWall = [sum(i) for i in zip(*dictionary_ScP_InteriorWall)]
GWP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[0])
ODP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[1])
POCP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[2])
AP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[3])
EP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[4])
HTP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[5])
FAETP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[6])
MAETP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[7])
TETP_ScP_InteriorWall = (dictionary_ScP_InteriorWall[8])
ADPc_ScP_InteriorWall = (dictionary_ScP_InteriorWall[9])
ADPf_ScP_InteriorWall = (dictionary_ScP_InteriorWall[10])
dictionary_ScP_InteriorWall = [GWP_ScP_InteriorWall, ODP_ScP_InteriorWall, POCP_ScP_InteriorWall, AP_ScP_InteriorWall,
EP_ScP_InteriorWall, HTP_ScP_InteriorWall, FAETP_ScP_InteriorWall, MAETP_ScP_InteriorWall, TETP_ScP_InteriorWall,
ADPc_ScP_InteriorWall, ADPf_ScP_InteriorWall]

TotalSCp_InteriorWall = sum(dictionary_ScP_InteriorWall)

### Floorcode 23###
importlib.reload(EC_2301003)
importlib.reload(EC_2301004)
importlib.reload(EC_2301012)
importlib.reload(EC_2301015)
importlib.reload(EC_2301072)

if building_system == "All":
    dictionary_MEFp_p_Floor = [EC_2301003.dictionary_MEFp_p, EC_2301015.dictionary_MEFp_p, EC_2301004.dictionary_MEFp_p,
    EC_2301012.dictionary_MEFp_p, EC_2301072.dictionary_MEFp_p]
    dictionary_MEFp_g_Floor = EC_2301003.dictionary_MEFp_p + EC_2301015.dictionary_MEFp_p + EC_2301004.dictionary_MEFp_p+
    EC_2301012.dictionary_MEFp_p+ EC_2301072.dictionary_MEFp_p
    dictionary_MEFp_s_Floor = EC_2301003.dictionary_MEFp_s, EC_2301015.dictionary_MEFp_s, EC_2301004.dictionary_MEFp_s,
    EC_2301012.dictionary_MEFp_s, EC_2301072.dictionary_MEFp_s

    dictionary_ScP_p_Floor = [EC_2301003.dictionary_ScP_p, EC_2301015.dictionary_ScP_p, EC_2301004.dictionary_ScP_p,
    EC_2301012.dictionary_ScP_p, EC_2301072.dictionary_ScP_p]
    dictionary_ScP_g_Floor = EC_2301003.dictionary_ScP_p + EC_2301015.dictionary_ScP_p+ EC_2301004.dictionary_ScP_p+
    EC_2301012.dictionary_ScP_p+ EC_2301072.dictionary_ScP_p
    dictionary_ScP_s_Floor = EC_2301003.dictionary_ScP_s, EC_2301015.dictionary_ScP_s, EC_2301004.dictionary_ScP_s,
    EC_2301012.dictionary_ScP_s, EC_2301072.dictionary_ScP_s
    dictionary_ScP_tot_g_Floor = sum(EC_2301003.dictionary_ScP_p), sum(EC_2301015.dictionary_ScP_p),
    sum(EC_2301004.dictionary_ScP_p), sum(EC_2301012.dictionary_ScP_p), sum(EC_2301072.dictionary_ScP_p)

    dictionary_Type_list_Floor = [EC_2301003.Type, EC_2301015.Type, EC_2301004.Type, EC_2301012.Type, EC_2301072.Type]
    dictionary_CostFloor_p_Floor = list(map(int, (EC_2301003.Cost_tot_p, EC_2301015.Cost_tot_p, EC_2301004.Cost_tot_p,

```

```

EC_2301012.Cost_tot_p, EC_2301072.Cost_tot_p)))
dictionary_CostFloor_g_Floor = EC_2301003.Cost_tot_p+ EC_2301015.Cost_tot_p+ EC_2301004.Cost_tot_p+ EC_2301012.Cost_tot_p+
EC_2301072.Cost_tot_p
dictionary_CostFloor_s_Floor = EC_2301003.Cost_tot_s, EC_2301015.Cost_tot_s, EC_2301004.Cost_tot_s, EC_2301012.Cost_tot_s,
EC_2301072.Cost_tot_s
dictionary_SurfaceFloor_g_Floor = EC_2301003.surface_area_gen + EC_2301015.surface_area_gen + EC_2301004.surface_area_gen +
EC_2301012.surface_area_gen + EC_2301072.surface_area_gen
dictionary_SurfaceFloor_g_Floor_ad = 0
dictionary_SurfaceFloor_s_Floor = EC_2301003.surface_area_spe + EC_2301015.surface_area_spe + EC_2301004.surface_area_spe +
EC_2301012.surface_area_spe + EC_2301072.surface_area_spe
elif building_system == "Bricks and concrete":
    dictionary_MEFp_p_Floor = [EC_2301003.dictionary_MEFp_p, EC_2301004.dictionary_MEFp_p, EC_2301012.dictionary_MEFp_p,
    EC_2301072.dictionary_MEFp_p]
    dictionary_MEFp_g_Floor = EC_2301003.dictionary_MEFp_p + EC_2301004.dictionary_MEFp_p+ EC_2301012.dictionary_MEFp_p+
    EC_2301072.dictionary_MEFp_p
    dictionary_MEFp_s_Floor = EC_2301003.dictionary_MEFp_s, EC_2301004.dictionary_MEFp_s, EC_2301012.dictionary_MEFp_s,
    EC_2301072.dictionary_MEFp_s

    dictionary_SCp_p_Floor = [EC_2301003.dictionary_SCp_p, EC_2301004.dictionary_SCp_p, EC_2301012.dictionary_SCp_p,
    EC_2301072.dictionary_SCp_p]
    dictionary_SCp_g_Floor = EC_2301003.dictionary_SCp_p + EC_2301004.dictionary_SCp_p+ EC_2301012.dictionary_SCp_p+
    EC_2301072.dictionary_SCp_p
    dictionary_SCp_s_Floor = EC_2301003.dictionary_SCp_s, EC_2301004.dictionary_SCp_s, EC_2301012.dictionary_SCp_s,
    EC_2301072.dictionary_SCp_s
    dictionary_SCp_tot_g_Floor = sum(EC_2301003.dictionary_SCp_p), sum(EC_2301004.dictionary_SCp_p),
    sum(EC_2301012.dictionary_SCp_p), sum(EC_2301072.dictionary_SCp_p)

    dictionary_Type_list_Floor = [EC_2301003.Type, EC_2301004.Type, EC_2301012.Type, EC_2301072.Type]
    dictionary_CostFloor_p_Floor = list(map(int, (EC_2301003.Cost_tot_p, EC_2301004.Cost_tot_p, EC_2301012.Cost_tot_p,
    EC_2301072.Cost_tot_p)))
    dictionary_CostFloor_g_Floor = EC_2301003.Cost_tot_p+ EC_2301004.Cost_tot_p+ EC_2301012.Cost_tot_p+ EC_2301072.Cost_tot_p
    dictionary_CostFloor_s_Floor = EC_2301003.Cost_tot_s, EC_2301004.Cost_tot_s, EC_2301012.Cost_tot_s, EC_2301072.Cost_tot_s
    dictionary_SurfaceFloor_g_Floor = EC_2301003.surface_area_gen + EC_2301004.surface_area_gen + EC_2301012.surface_area_gen +
    EC_2301072.surface_area_gen
    dictionary_SurfaceFloor_g_Floor_ad = EC_2301015.surface_area_spe
    dictionary_SurfaceFloor_s_Floor = EC_2301003.surface_area_spe + EC_2301004.surface_area_spe + EC_2301012.surface_area_spe +
    EC_2301072.surface_area_spe
elif building_system == "Wood":
    dictionary_MEFp_p_Floor = [EC_2301015.dictionary_MEFp_p]
    dictionary_MEFp_g_Floor = EC_2301015.dictionary_MEFp_p
    dictionary_MEFp_s_Floor = dictionary_empty_start, EC_2301015.dictionary_MEFp_s

    dictionary_SCp_p_Floor = [EC_2301015.dictionary_SCp_p]
    dictionary_SCp_g_Floor = EC_2301015.dictionary_SCp_p
    dictionary_SCp_s_Floor = dictionary_empty_start, EC_2301015.dictionary_SCp_s
    dictionary_SCp_tot_g_Floor = sum(EC_2301015.dictionary_SCp_p)

    dictionary_Type_list_Floor = [EC_2301015.Type]
    dictionary_CostFloor_p_Floor = [EC_2301015.Cost_tot_p]
    dictionary_CostFloor_g_Floor = EC_2301015.Cost_tot_p
    dictionary_CostFloor_s_Floor = EC_2301015.Cost_tot_s
    dictionary_SurfaceFloor_g_Floor = EC_2301015.surface_area_gen
    dictionary_SurfaceFloor_g_Floor_ad = EC_2301003.surface_area_spe + EC_2301004.surface_area_spe + EC_2301012.surface_area_spe +
    EC_2301072.surface_area_spe
    dictionary_SurfaceFloor_s_Floor = EC_2301015.surface_area_spe

countofelements_Floor = len(dictionary_MEFp_p_Floor)

if dictionary_SurfaceFloor_g_Floor < 0:
    dictionary_SurfaceFloor_g_Floor = 0
else:
    dictionary_SurfaceFloor_g_Floor = dictionary_SurfaceFloor_g_Floor / int(countofelements_Floor)

if calculation_choice == "Generic":
    FloorSurfaceGeneric_Floor = dictionary_SurfaceFloor_g_Floor + dictionary_SurfaceFloor_g_Floor_ad + dictionary_SurfaceFloor_s_Floor
    FloorSurfaceSpecific_Floor = 0
    CostFloorGeneric_Floor = dictionary_CostFloor_g_Floor / int(countofelements_Floor) * FloorSurfaceGeneric_Floor
    CostFloorSpecific_Floor = 0
elif calculation_choice == "Specific":
    FloorSurfaceGeneric_Floor = dictionary_SurfaceFloor_g_Floor + dictionary_SurfaceFloor_g_Floor_ad
    FloorSurfaceSpecific_Floor = dictionary_SurfaceFloor_s_Floor

```

```

CostFloorGeneric_Floor = dictionary_CostFloor_g_Floor / int(countofelements_Floor) * FloorSurfaceGeneric_Floor
if countofelements_Floor == 1:
    CostFloorSpecific_Floor = dictionary_CostFloor_s_Floor
elif countofelements_Floor > 1:
    CostFloorSpecific_Floor = sum(dictionary_CostFloor_s_Floor)
else:
    FloorSurfaceGeneric_Floor = 0
    FloorSurfaceSpecific_Floor = 0
    CostFloorGeneric_Floor = 0
    CostFloorSpecific_Floor = 0

dictionary_MEFP_g_Floor = [FloorSurfaceGeneric_Floor * i for i in dictionary_MEFP_g_Floor]
dictionary_SCp_g_Floor = [FloorSurfaceGeneric_Floor * i for i in dictionary_SCp_g_Floor]
dictionary_CostFloor_g_Floor = [FloorSurfaceGeneric_Floor * i for i in dictionary_CostFloor_p_Floor]

def divide_list(l, n):
    for i in range(0, len(l), n):
        yield l[i:i + n]
n = 11
dictionary_MEFP_g_Floor = list(divide_list(dictionary_MEFP_g_Floor, n))
dictionary_SCp_g_Floor = list(divide_list(dictionary_SCp_g_Floor, n))

bestchoice_Floor_test = FloorSurfaceGeneric_Floor
if countofelements_Floor == 1:
    if bestchoice_Floor_test > 0:
        bestchoice_envimp_Floor_type = dictionary_Type_list_Floor[0]
        bestchoice_ecoimp_Floor_type = dictionary_Type_list_Floor[0]
        bestchoice_overall_Floor_type = dictionary_Type_list_Floor[0]
    elif bestchoice_Floor_test > 0:
        dictionary_SCp_total_g_Floor = [FloorSurfaceGeneric_Floor * i for i in dictionary_SCp_tot_g_Floor]
        bestchoice_envimp_Floor_np = np.array(dictionary_SCp_total_g_Floor)
        bestchoice_envimp_Floor_nrlist_min = bestchoice_envimp_Floor_np.argmin()
        bestchoice_envimp_Floor_nrlist_max = bestchoice_envimp_Floor_np.argmax()
        bestchoice_envimp_Floor_type = dictionary_Type_list_Floor[bestchoice_envimp_Floor_nrlist_min]
        bestchoice_envimp_Floor = dictionary_SCp_total_g_Floor[bestchoice_envimp_Floor_nrlist_min]
        bestchoice_envimp_Floor_max = dictionary_SCp_total_g_Floor[bestchoice_envimp_Floor_nrlist_max]
        bestchoice_ecoimp_Floor_np = np.array(dictionary_CostFloor_g_Floor)
        bestchoice_ecoimp_Floor_nrlist_min = bestchoice_ecoimp_Floor_np.argmin()
        bestchoice_ecoimp_Floor_nrlist_max = bestchoice_ecoimp_Floor_np.argmax()
        bestchoice_ecoimp_Floor_type = dictionary_Type_list_Floor[bestchoice_ecoimp_Floor_nrlist_min]
        bestchoice_ecoimp_Floor = dictionary_CostFloor_g_Floor[bestchoice_ecoimp_Floor_nrlist_min]
        bestchoice_ecoimp_Floor_max = dictionary_CostFloor_g_Floor[bestchoice_ecoimp_Floor_nrlist_max]
        SCp_g_list_Floor_optimal = [x / bestchoice_envimp_Floor_max * factorenv for x in dictionary_SCp_total_g_Floor]
        Cost_g_list_Floor_optimal = [x / bestchoice_ecoimp_Floor_max * factoreco for x in dictionary_CostFloor_g_Floor]
        Overall_list_Floor = {"Cost": Cost_g_list_Floor_optimal, "Environment": SCp_g_list_Floor_optimal}
        Overall_list_Floor_optimal = []
        for value in (zip(*list(Overall_list_Floor.values()))):
            Overall_list_Floor_optimal.append(sum(value))
        linearprogramming_Floor = np.array(Overall_list_Floor_optimal)
        bestchoice_overall_Floor_nrlist = linearprogramming_Floor.argmin()
        bestchoice_overall_Floor_type = dictionary_Type_list_Floor[bestchoice_overall_Floor_nrlist]
    else:
        bestchoice_envimp_Floor_type = "-"
        bestchoice_ecoimp_Floor_type = "-"
        bestchoice_overall_Floor_type = "-"

dictionary_MEFP_p_Floor = [sum(i) for i in zip(*dictionary_MEFP_p_Floor)]
GWP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[0]) / countofelements_Floor * FloorSurfaceGeneric_Floor
ODP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[1]) / countofelements_Floor * FloorSurfaceGeneric_Floor
POCP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[2]) / countofelements_Floor * FloorSurfaceGeneric_Floor
AP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[3]) / countofelements_Floor * FloorSurfaceGeneric_Floor
EP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[4]) / countofelements_Floor * FloorSurfaceGeneric_Floor
HTP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[5]) / countofelements_Floor * FloorSurfaceGeneric_Floor
FAETP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[6]) / countofelements_Floor * FloorSurfaceGeneric_Floor
MAETP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[7]) / countofelements_Floor * FloorSurfaceGeneric_Floor
TETP_MEFP_g_Floor = (dictionary_MEFP_p_Floor[8]) / countofelements_Floor * FloorSurfaceGeneric_Floor
ADPc_MEFP_g_Floor = (dictionary_MEFP_p_Floor[9]) / countofelements_Floor * FloorSurfaceGeneric_Floor
ADPf_MEFP_g_Floor = (dictionary_MEFP_p_Floor[10]) / countofelements_Floor * FloorSurfaceGeneric_Floor
dictionary_MEFP_g_Floor_average = [GWP_MEFP_g_Floor, ODP_MEFP_g_Floor, POCP_MEFP_g_Floor, AP_MEFP_g_Floor,
EP_MEFP_g_Floor, HTP_MEFP_g_Floor, FAETP_MEFP_g_Floor, MAETP_MEFP_g_Floor, TETP_MEFP_g_Floor, ADPc_MEFP_g_Floor,
ADPf_MEFP_g_Floor]

```

```

dictionary_MEFp_s_Floor = [sum(i) for i in zip(*dictionary_MEFp_s_Floor)]
dictionary_MEFp_Floor = dictionary_MEFp_s_Floor, dictionary_MEFp_g_Floor_average
dictionary_MEFp_Floor = [sum(i) for i in zip(*dictionary_MEFp_Floor)]
GWP_MEFp_Floor = (dictionary_MEFp_Floor[0])
ODP_MEFp_Floor = (dictionary_MEFp_Floor[1])
POCP_MEFp_Floor = (dictionary_MEFp_Floor[2])
AP_MEFp_Floor = (dictionary_MEFp_Floor[3])
EP_MEFp_Floor = (dictionary_MEFp_Floor[4])
HTP_MEFp_Floor = (dictionary_MEFp_Floor[5])
FAETP_MEFp_Floor = (dictionary_MEFp_Floor[6])
MAETP_MEFp_Floor = (dictionary_MEFp_Floor[7])
TETP_MEFp_Floor = (dictionary_MEFp_Floor[8])
ADPc_MEFp_Floor = (dictionary_MEFp_Floor[9])
ADPf_MEFp_Floor = (dictionary_MEFp_Floor[10])

dictionary_SCp_p_Floor = [sum(i) for i in zip(*dictionary_SCp_p_Floor)]
GWP_SCp_g_Floor = (dictionary_SCp_p_Floor[0]) / countofelements_Floor * FloorSurfaceGeneric_Floor
ODP_SCp_g_Floor = (dictionary_SCp_p_Floor[1]) / countofelements_Floor * FloorSurfaceGeneric_Floor
POCP_SCp_g_Floor = (dictionary_SCp_p_Floor[2]) / countofelements_Floor * FloorSurfaceGeneric_Floor
AP_SCp_g_Floor = (dictionary_SCp_p_Floor[3]) / countofelements_Floor * FloorSurfaceGeneric_Floor
EP_SCp_g_Floor = (dictionary_SCp_p_Floor[4]) / countofelements_Floor * FloorSurfaceGeneric_Floor
HTP_SCp_g_Floor = (dictionary_SCp_p_Floor[5]) / countofelements_Floor * FloorSurfaceGeneric_Floor
FAETP_SCp_g_Floor = (dictionary_SCp_p_Floor[6]) / countofelements_Floor * FloorSurfaceGeneric_Floor
MAETP_SCp_g_Floor = (dictionary_SCp_p_Floor[7]) / countofelements_Floor * FloorSurfaceGeneric_Floor
TETP_SCp_g_Floor = (dictionary_SCp_p_Floor[8]) / countofelements_Floor * FloorSurfaceGeneric_Floor
ADPc_SCp_g_Floor = (dictionary_SCp_p_Floor[9]) / countofelements_Floor * FloorSurfaceGeneric_Floor
ADPf_SCp_g_Floor = (dictionary_SCp_p_Floor[10]) / countofelements_Floor * FloorSurfaceGeneric_Floor
dictionary_SCp_g_Floor_average = [GWP_SCp_g_Floor, ODP_SCp_g_Floor, POCP_SCp_g_Floor, AP_SCp_g_Floor, EP_SCp_g_Floor,
HTP_SCp_g_Floor, FAETP_SCp_g_Floor, MAETP_SCp_g_Floor, TETP_SCp_g_Floor, ADPc_SCp_g_Floor, ADPf_SCp_g_Floor]

dictionary_SCp_g_Floor_totals = (list(sum(i) for i in dictionary_SCp_g_Floor))
dictionary_SCp_g_Floor_totals = [round(x,2) for x in dictionary_SCp_g_Floor_totals]
dictionary_SCp_s_Floor_totals = (list(sum(i) for i in dictionary_SCp_s_Floor))
dictionary_SCp_s_Floor_totals = [round(x,2) for x in dictionary_SCp_s_Floor_totals]

dictionary_SCp_s_Floor = [sum(i) for i in zip(*dictionary_SCp_s_Floor)]
dictionary_SCp_Floor = dictionary_SCp_s_Floor, dictionary_SCp_g_Floor_average
dictionary_SCp_Floor = [sum(i) for i in zip(*dictionary_SCp_Floor)]
GWP_SCp_Floor = (dictionary_SCp_Floor[0])
ODP_SCp_Floor = (dictionary_SCp_Floor[1])
POCP_SCp_Floor = (dictionary_SCp_Floor[2])
AP_SCp_Floor = (dictionary_SCp_Floor[3])
EP_SCp_Floor = (dictionary_SCp_Floor[4])
HTP_SCp_Floor = (dictionary_SCp_Floor[5])
FAETP_SCp_Floor = (dictionary_SCp_Floor[6])
MAETP_SCp_Floor = (dictionary_SCp_Floor[7])
TETP_SCp_Floor = (dictionary_SCp_Floor[8])
ADPc_SCp_Floor = (dictionary_SCp_Floor[9])
ADPf_SCp_Floor = (dictionary_SCp_Floor[10])
dictionary_SCp_Floor = [GWP_SCp_Floor, ODP_SCp_Floor, POCP_SCp_Floor, AP_SCp_Floor, EP_SCp_Floor, HTP_SCp_Floor,
FAETP_SCp_Floor, MAETP_SCp_Floor, TETP_SCp_Floor, ADPc_SCp_Floor, ADPf_SCp_Floor]
TotalSCp_Floor = sum(dictionary_SCp_Floor)

### Wallcode 28###
importlib.reload(EC_2801003)
importlib.reload(EC_2801028)
importlib.reload(EC_2801029)

if building_system == "All":
    dictionary_MEFp_p_ExteriorWall = [EC_2801003.dictionary_MEFp_p, EC_2801028.dictionary_MEFp_p, EC_2801029.dictionary_MEFp_p]
    dictionary_MEFp_g_ExteriorWall = EC_2801003.dictionary_MEFp_p + EC_2801028.dictionary_MEFp_p + EC_2801029.dictionary_MEFp_p
    dictionary_MEFp_s_ExteriorWall = EC_2801003.dictionary_MEFp_s, EC_2801028.dictionary_MEFp_s, EC_2801029.dictionary_MEFp_s

    dictionary_SCp_p_ExteriorWall = [EC_2801003.dictionary_SCp_p, EC_2801028.dictionary_SCp_p, EC_2801029.dictionary_SCp_p]
    dictionary_SCp_g_ExteriorWall = EC_2801003.dictionary_SCp_p + EC_2801028.dictionary_SCp_p + EC_2801029.dictionary_SCp_p
    dictionary_SCp_s_ExteriorWall = EC_2801003.dictionary_SCp_s, EC_2801028.dictionary_SCp_s, EC_2801029.dictionary_SCp_s
    dictionary_SCp_tot_g_ExteriorWall = sum(EC_2801003.dictionary_SCp_p), sum(EC_2801028.dictionary_SCp_p),
sum(EC_2801029.dictionary_SCp_p)

    dictionary_Type_list_ExteriorWall = [EC_2801003.Type, EC_2801028.Type, EC_2801029.Type]

```

```

dictionary_CostWall_p_ExteriorWall = list(map(int, (EC_2801003.Cost_tot_p, EC_2801028.Cost_tot_p, EC_2801029.Cost_tot_p)))
dictionary_CostWall_g_ExteriorWall = EC_2801003.Cost_tot_p + EC_2801028.Cost_tot_p + EC_2801029.Cost_tot_p
dictionary_CostWall_s_ExteriorWall = EC_2801003.Cost_tot_s, EC_2801028.Cost_tot_s, EC_2801029.Cost_tot_s
dictionary_SurfaceWall_g_ExteriorWall = EC_2801003.surface_area_gen + EC_2801028.surface_area_gen +
EC_2801029.surface_area_gen
dictionary_SurfaceWall_g_ExteriorWall_ad = 0
dictionary_SurfaceWall_s_ExteriorWall = EC_2801003.surface_area_spe + EC_2801028.surface_area_spe + EC_2801029.surface_area_spe

elif building_system == "Bricks and concrete":
dictionary_MEFp_p_ExteriorWall = [EC_2801003.dictionary_MEFp_p, EC_2801029.dictionary_MEFp_p]
dictionary_MEFp_g_ExteriorWall = EC_2801003.dictionary_MEFp_p + EC_2801029.dictionary_MEFp_p
dictionary_MEFp_s_ExteriorWall = EC_2801003.dictionary_MEFp_s, EC_2801029.dictionary_MEFp_s

dictionary_SCp_p_ExteriorWall = [EC_2801003.dictionary_SCp_p, EC_2801029.dictionary_SCp_p]
dictionary_SCp_g_ExteriorWall = EC_2801003.dictionary_SCp_p + EC_2801029.dictionary_SCp_p
dictionary_SCp_s_ExteriorWall = EC_2801003.dictionary_SCp_s, EC_2801029.dictionary_SCp_s
dictionary_SCp_tot_g_ExteriorWall = sum(EC_2801003.dictionary_SCp_p), sum(EC_2801029.dictionary_SCp_p)

dictionary_Type_list_ExteriorWall = [EC_2801003.Type, EC_2801029.Type]
dictionary_CostWall_p_ExteriorWall = list(map(int, (EC_2801003.Cost_tot_p, EC_2801029.Cost_tot_p)))
dictionary_CostWall_g_ExteriorWall = EC_2801003.Cost_tot_p + EC_2801029.Cost_tot_p
dictionary_CostWall_s_ExteriorWall = EC_2801003.Cost_tot_s, EC_2801029.Cost_tot_s
dictionary_SurfaceWall_g_ExteriorWall = EC_2801003.surface_area_gen + EC_2801029.surface_area_gen
dictionary_SurfaceWall_g_ExteriorWall_ad = EC_2801028.surface_area_spe
dictionary_SurfaceWall_s_ExteriorWall = EC_2801003.surface_area_spe + EC_2801029.surface_area_spe

elif building_system == "Wood":
dictionary_MEFp_p_ExteriorWall = [EC_2801028.dictionary_MEFp_p]
dictionary_MEFp_g_ExteriorWall = EC_2801028.dictionary_MEFp_p
dictionary_MEFp_s_ExteriorWall = dictionary_empty_start, EC_2801028.dictionary_MEFp_s

dictionary_SCp_p_ExteriorWall = [EC_2801028.dictionary_SCp_p]
dictionary_SCp_g_ExteriorWall = EC_2801028.dictionary_SCp_p
dictionary_SCp_s_ExteriorWall = dictionary_empty_start, EC_2801028.dictionary_SCp_s
dictionary_SCp_tot_g_ExteriorWall = sum(EC_2803001.dictionary_SCp_p)

dictionary_Type_list_ExteriorWall = [EC_2801028.Type]
dictionary_CostWall_p_ExteriorWall = [EC_2801028.Cost_tot_p]
dictionary_CostWall_g_ExteriorWall = EC_2801028.Cost_tot_p
dictionary_CostWall_s_ExteriorWall = EC_2801028.Cost_tot_s
dictionary_SurfaceWall_g_ExteriorWall = EC_2801028.surface_area_gen
dictionary_SurfaceWall_g_ExteriorWall_ad = EC_2801003.surface_area_spe + EC_2801029.surface_area_spe
dictionary_SurfaceWall_s_ExteriorWall = EC_2801028.surface_area_spe

countofelements_ExteriorWall = len(dictionary_MEFp_p_ExteriorWall)

if dictionary_SurfaceWall_g_ExteriorWall < 0:
dictionary_SurfaceWall_g_ExteriorWall = 0
else:
dictionary_SurfaceWall_g_ExteriorWall = dictionary_SurfaceWall_g_ExteriorWall / int(countofelements_ExteriorWall)

if calculation_choice == "Generic":
WallSurfaceGeneric_ExteriorWall = dictionary_SurfaceWall_g_ExteriorWall + dictionary_SurfaceWall_g_ExteriorWall_ad +
dictionary_SurfaceWall_s_ExteriorWall
WallSurfaceSpecific_ExteriorWall = 0
CostWallGeneric_ExteriorWall = dictionary_CostWall_g_ExteriorWall / int(countofelements_ExteriorWall) *
WallSurfaceGeneric_ExteriorWall
CostWallSpecific_ExteriorWall = 0
elif calculation_choice == "Specific":
WallSurfaceGeneric_ExteriorWall = dictionary_SurfaceWall_g_ExteriorWall + dictionary_SurfaceWall_g_ExteriorWall_ad
WallSurfaceSpecific_ExteriorWall = dictionary_SurfaceWall_s_ExteriorWall
CostWallGeneric_ExteriorWall = dictionary_CostWall_g_ExteriorWall / int(countofelements_ExteriorWall) *
WallSurfaceGeneric_ExteriorWall
if countofelements_InteriorWall == 1:
CostWallSpecific_ExteriorWall = dictionary_CostWall_s_ExteriorWall
elif countofelements_InteriorWall > 1:
CostWallSpecific_ExteriorWall = sum(dictionary_CostWall_s_ExteriorWall)
else:
WallSurfaceGeneric_ExteriorWall = 0
WallSurfaceSpecific_ExteriorWall = 0
CostWallGeneric_ExteriorWall = 0

```

```

CostWallSpecific_ExteriorWall = 0

dictionary_MEFp_g_ExteriorWall = [WallSurfaceGeneric_ExteriorWall * i for i in dictionary_MEFp_g_ExteriorWall]
dictionary_SCp_g_ExteriorWall = [WallSurfaceGeneric_ExteriorWall * i for i in dictionary_SCp_g_ExteriorWall]
dictionary_CostWall_g_ExteriorWall = [WallSurfaceGeneric_ExteriorWall * i for i in dictionary_CostWall_p_ExteriorWall]

def divide_list_exteriorwall(l, n):
    for i in range(0, len(l), n):
        yield l[i:i + n]
n = 11
dictionary_MEFp_g_ExteriorWall = list(divide_list_exteriorwall(dictionary_MEFp_g_ExteriorWall, n))
dictionary_SCp_g_ExteriorWall = list(divide_list_exteriorwall(dictionary_SCp_g_ExteriorWall, n))

bestchoice_ExteriorWall_test = WallSurfaceGeneric_ExteriorWall
if countofelements_ExteriorWall == 1:
    if bestchoice_ExteriorWall_test > 0:
        bestchoice_envimp_ExteriorWall_type = dictionary_Type_list_ExteriorWall[0]
        bestchoice_ecoimp_ExteriorWall_type = dictionary_Type_list_ExteriorWall[0]
        bestchoice_overall_ExteriorWall_type = dictionary_Type_list_ExteriorWall[0]
    elif bestchoice_ExteriorWall_test > 0:
        dictionary_SCp_total_g_ExteriorWall = [WallSurfaceGeneric_ExteriorWall * i for i in dictionary_SCp_tot_g_ExteriorWall]
        bestchoice_envimp_ExteriorWall_np = np.array(dictionary_SCp_total_g_ExteriorWall)
        bestchoice_envimp_ExteriorWall_nrlist_min = bestchoice_envimp_ExteriorWall_np.argmin()
        bestchoice_envimp_ExteriorWall_nrlist_max = bestchoice_envimp_ExteriorWall_np.argmax()
        bestchoice_envimp_ExteriorWall_type = dictionary_Type_list_ExteriorWall[bestchoice_envimp_ExteriorWall_nrlist_min]
        bestchoice_envimp_ExteriorWall = dictionary_SCp_total_g_ExteriorWall[bestchoice_envimp_ExteriorWall_nrlist_min]
        bestchoice_envimp_ExteriorWall_max = dictionary_SCp_total_g_ExteriorWall[bestchoice_envimp_ExteriorWall_nrlist_max]
        bestchoice_ecoimp_ExteriorWall_np = np.array(dictionary_CostWall_g_ExteriorWall)
        bestchoice_ecoimp_ExteriorWall_nrlist_min = bestchoice_ecoimp_ExteriorWall_np.argmin()
        bestchoice_ecoimp_ExteriorWall_nrlist_max = bestchoice_ecoimp_ExteriorWall_np.argmax()
        bestchoice_ecoimp_ExteriorWall_type = dictionary_Type_list_ExteriorWall[bestchoice_ecoimp_ExteriorWall_nrlist_min]
        bestchoice_ecoimp_ExteriorWall = dictionary_CostWall_g_ExteriorWall[bestchoice_ecoimp_ExteriorWall_nrlist_min]
        bestchoice_ecoimp_ExteriorWall_max = dictionary_CostWall_g_ExteriorWall[bestchoice_ecoimp_ExteriorWall_nrlist_max]
        SCp_g_list_ExteriorWall_optimal = [x / bestchoice_envimp_ExteriorWall_max * factorenv for x in dictionary_SCp_total_g_ExteriorWall]
        Cost_g_list_ExteriorWall_optimal = [x / bestchoice_ecoimp_ExteriorWall_max * factoreco for x in dictionary_CostWall_g_ExteriorWall]
        Overall_list_ExteriorWall = {"Cost": Cost_g_list_ExteriorWall_optimal, "Environment": SCp_g_list_ExteriorWall_optimal}
        Overall_list_ExteriorWall_optimal = []
        for value in (zip(*list(Overall_list_ExteriorWall.values()))):
            Overall_list_ExteriorWall_optimal.append(sum(value))
        linearprogramming_ExteriorWall = np.array(Overall_list_ExteriorWall_optimal)
        bestchoice_overall_ExteriorWall_nrlist = linearprogramming_ExteriorWall.argmin()
        bestchoice_overall_ExteriorWall_type = dictionary_Type_list_ExteriorWall[bestchoice_overall_ExteriorWall_nrlist]
    else:
        bestchoice_envimp_ExteriorWall_type = "-"
        bestchoice_ecoimp_ExteriorWall_type = "-"
        bestchoice_overall_ExteriorWall_type = "-"

dictionary_MEFp_p_ExteriorWall = [sum(i) for i in zip(*dictionary_MEFp_p_ExteriorWall)]
GWP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[0]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
ODP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[1]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
POCP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[2]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
AP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[3]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
EP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[4]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
HTP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[5]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
FAETP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[6]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
MAETP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[7]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
TETP_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[8]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
ADPc_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[9]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
ADPf_MEFp_g_ExteriorWall = (dictionary_MEFp_p_ExteriorWall[10]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
dictionary_MEFp_g_ExteriorWall_average = [GWP_MEFp_g_ExteriorWall, ODP_MEFp_g_ExteriorWall, POCP_MEFp_g_ExteriorWall,
AP_MEFp_g_ExteriorWall, EP_MEFp_g_ExteriorWall, HTP_MEFp_g_ExteriorWall, FAETP_MEFp_g_ExteriorWall,
MAETP_MEFp_g_ExteriorWall, TETP_MEFp_g_ExteriorWall, ADPc_MEFp_g_ExteriorWall, ADPf_MEFp_g_ExteriorWall]

dictionary_MEFp_s_ExteriorWall = [sum(i) for i in zip(*dictionary_MEFp_s_ExteriorWall)]
dictionary_MEFp_ExteriorWall = dictionary_MEFp_s_ExteriorWall, dictionary_MEFp_g_ExteriorWall_average
dictionary_MEFp_ExteriorWall = [sum(i) for i in zip(*dictionary_MEFp_ExteriorWall)]
GWP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[0])
ODP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[1])
POCP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[2])
AP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[3])
EP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[4])

```



```

HTP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[5])
FAETP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[6])
MAETP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[7])
TETP_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[8])
ADPc_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[9])
ADPf_MEFp_ExteriorWall = (dictionary_MEFp_ExteriorWall[10])

dictionary_SCp_p_ExteriorWall = [sum(i) for i in zip(*dictionary_SCp_p_ExteriorWall)]
GWP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[0]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
ODP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[1]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
POCP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[2]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
AP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[3]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
EP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[4]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
HTP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[5]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
FAETP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[6]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
MAETP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[7]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
TETP_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[8]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
ADPc_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[9]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
ADPf_SCp_g_ExteriorWall = (dictionary_SCp_p_ExteriorWall[10]) / countofelements_ExteriorWall * WallSurfaceGeneric_ExteriorWall
dictionary_SCp_g_ExteriorWall_average = [GWP_SCp_g_ExteriorWall, ODP_SCp_g_ExteriorWall, POCP_SCp_g_ExteriorWall, AP_SCp_g_ExteriorWall, EP_SCp_g_ExteriorWall, HTP_SCp_g_ExteriorWall, FAETP_SCp_g_ExteriorWall, MAETP_SCp_g_ExteriorWall, TETP_SCp_g_ExteriorWall, ADPc_SCp_g_ExteriorWall, ADPf_SCp_g_ExteriorWall]

dictionary_SCp_g_ExteriorWall_totals = (list(sum(i) for i in dictionary_SCp_g_ExteriorWall))
dictionary_SCp_g_ExteriorWall_totals = [round(x,2) for x in dictionary_SCp_g_ExteriorWall_totals]
dictionary_SCp_s_ExteriorWall_totals = (list(sum(i) for i in dictionary_SCp_s_ExteriorWall))
dictionary_SCp_s_ExteriorWall_totals = [round(x,2) for x in dictionary_SCp_s_ExteriorWall_totals]

dictionary_SCp_s_ExteriorWall = [sum(i) for i in zip(*dictionary_SCp_s_ExteriorWall)]
dictionary_SCp_ExteriorWall = dictionary_SCp_s_ExteriorWall, dictionary_SCp_g_ExteriorWall_average
dictionary_SCp_ExteriorWall = [sum(i) for i in zip(*dictionary_SCp_ExteriorWall)]
GWP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[0])
ODP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[1])
POCP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[2])
AP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[3])
EP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[4])
HTP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[5])
FAETP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[6])
MAETP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[7])
TETP_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[8])
ADPc_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[9])
ADPf_SCp_ExteriorWall = (dictionary_SCp_ExteriorWall[10])
dictionary_SCp_ExteriorWall = [GWP_SCp_ExteriorWall, ODP_SCp_ExteriorWall, POCP_SCp_ExteriorWall, AP_SCp_ExteriorWall, EP_SCp_ExteriorWall, HTP_SCp_ExteriorWall, FAETP_SCp_ExteriorWall, MAETP_SCp_ExteriorWall, TETP_SCp_ExteriorWall, ADPc_SCp_ExteriorWall, ADPf_SCp_ExteriorWall]
TotalSCp_ExteriorWall = sum(dictionary_SCp_ExteriorWall)

###OVERALL
dictionary_MEFp_sum = dictionary_MEFp_InteriorWall, dictionary_MEFp_Floor, dictionary_MEFp_ExteriorWall
dictionary_MEFp_sum = list(zip(*dictionary_MEFp_sum))
GWP_MEFp = sum(dictionary_MEFp_sum[0])
ODP_MEFp = sum(dictionary_MEFp_sum[1])
POCP_MEFp = sum(dictionary_MEFp_sum[2])
AP_MEFp = sum(dictionary_MEFp_sum[3])
EP_MEFp = sum(dictionary_MEFp_sum[4])
HTP_MEFp = sum(dictionary_MEFp_sum[5])
FAETP_MEFp = sum(dictionary_MEFp_sum[6])
MAETP_MEFp = sum(dictionary_MEFp_sum[7])
TETP_MEFp = sum(dictionary_MEFp_sum[8])
ADPc_MEFp = sum(dictionary_MEFp_sum[9])
ADPf_MEFp = sum(dictionary_MEFp_sum[10])

dictionary_SCp_sum = dictionary_SCp_InteriorWall, dictionary_SCp_Floor, dictionary_SCp_ExteriorWall
dictionary_SCp_sum = list(zip(*dictionary_SCp_sum))
GWP_SCp = sum(dictionary_SCp_sum[0])
ODP_SCp = sum(dictionary_SCp_sum[1])
POCP_SCp = sum(dictionary_SCp_sum[2])
AP_SCp = sum(dictionary_SCp_sum[3])
EP_SCp = sum(dictionary_SCp_sum[4])
HTP_SCp = sum(dictionary_SCp_sum[5])
FAETP_SCp = sum(dictionary_SCp_sum[6])

```



```
MAETP_SCp = sum(dictionary_SCp_sum[7])
TETP_SCp = sum(dictionary_SCp_sum[8])
ADPc_SCp = sum(dictionary_SCp_sum[9])
ADPf_SCp = sum(dictionary_SCp_sum[10])
```

###RESULTS

```
dictionary_MEFp = [GWP_MEFp, ODP_MEFp, POCP_MEFp, AP_MEFp, EP_MEFp, HTP_MEFp, FAETP_MEFp, MAETP_MEFp, TETP_MEFp,
ADPc_MEFp, ADPf_MEFp]
dictionary_SCp = [GWP_SCp, ODP_SCp, POCP_SCp, AP_SCp, EP_SCp, HTP_SCp, FAETP_SCp, MAETP_SCp, TETP_SCp, ADPc_SCp, ADPf_SCp]
TotalSCp = TotalSCp_InteriorWall + TotalSCp_Floor + TotalSCp_ExteriorWall
TotalCostGeneric = CostWallGeneric_InteriorWall + CostFloorGeneric_Floor + CostWallGeneric_ExteriorWall
TotalCostSpecific = CostWallSpecific_InteriorWall + CostFloorSpecific_Floor + CostWallSpecific_ExteriorWall
```

GUI.PY

```
# This Python file uses the following encoding: utf-8

from PyQt4 import QtCore, QtGui
from PyQt4.QtGui import QSlider

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1118, 846)
        sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Expanding)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightForWidth())
        MainWindow.setSizePolicy(sizePolicy)
        icon = QtGui.QIcon()
        icon.addPixmap(QtGui.QPixmap("../Library/img/logo.jpg"), QtGui.QIcon.Normal, QtGui.QIcon.Off)
        MainWindow.setWindowIcon(icon)
        self.mainapp = QtGui.QWidget(MainWindow)
        self.mainapp.setObjectName("mainapp")
        self.gridLayout = QtGui.QGridLayout(self.mainapp)
        self.gridLayout.setObjectName("gridLayout")
        self.layoutManualInput = QtGui.QVBoxLayout()
        self.layoutManualInput.setSizeConstraint(QtGui.QLayout.SetFixedSize)
        self.layoutManualInput.setObjectName("layoutManualInput")
        self.verticalLayout = QtGui.QVBoxLayout()
        self.verticalLayout.setObjectName("verticalLayout")
        spacerItem = QtGui.QSpacerItem(20, 50, QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Expanding)
        self.verticalLayout.addItem(spacerItem)
        self.labelInput = QtGui.QLabel(self.mainapp)
        sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Maximum)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(self.labelInput.sizePolicy().hasHeightForWidth())
        self.labelInput.setSizePolicy(sizePolicy)
        self.labelInput.setMinimumSize(QtCore.QSize(360, 18))
        self.labelInput.setMaximumSize(QtCore.QSize(16777215, 20))
        font = QtGui.QFont()
        font.setFamily("Calibri")
        font.setPointSize(11)
        font.setBold(True)
        font.setWeight(75)
        self.labelInput.setFont(font)
        self.labelInput.setFrameShape(QtGui.QFrame.NoFrame)
        self.labelInput.setLineWidth(0)
        self.labelInput.setScaledContents(False)
        self.labelInput.setAlignment(QtCore.Qt.AlignCenter)
        self.labelInput.setObjectName("labelInput")
        self.verticalLayout.addWidget(self.labelInput)
        self.line_22 = QtGui.QFrame(self.mainapp)
        font = QtGui.QFont()
        font.setFamily("Calibri")
        font.setPointSize(11)
        self.line_22.setFont(font)
        self.line_22.setFrameShadow(QtGui.QFrame.Sunken)
        self.line_22.setFrameShape(QtGui.QFrame.HLine)
        self.line_22.setObjectName("line_22")
        self.verticalLayout.addWidget(self.line_22)
        self.layoutManualInput.addLayout(self.verticalLayout)
        self.layoutBuildingType = QtGui.QHBoxLayout()
        self.layoutBuildingType.setObjectName("layoutBuildingType")
        self.labelBuildingType = QtGui.QLabel(self.mainapp)
        sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Maximum)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(self.labelBuildingType.sizePolicy().hasHeightForWidth())
        self.labelBuildingType.setSizePolicy(sizePolicy)
        self.labelBuildingType.setMinimumSize(QtCore.QSize(0, 17))
        self.labelBuildingType.setMaximumSize(QtCore.QSize(100, 16777215))
        font = QtGui.QFont()
```

```

font.setFamily("Calibri")
font.setPointSize(11)
self.labelBuildingType.setFont(font)
self.labelBuildingType.setObjectName("labelBuildingType")
self.layoutBuildingType.addWidget(self.labelBuildingType)
spacerItem1 = QtGui.QSpacerItem(105, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutBuildingType.addItem(spacerItem1)
self.buildingTypeChoice = QtGui.QComboBox(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Fixed)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.buildingTypeChoice.sizePolicy().hasHeightForWidth())
self.buildingTypeChoice.setSizePolicy(sizePolicy)
self.buildingTypeChoice.setMinimumSize(QtCore.QSize(150, 20))
self.buildingTypeChoice.setMaximumSize(QtCore.QSize(150, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.buildingTypeChoice.setFont(font)
self.buildingTypeChoice.setAutoFillBackground(False)
self.buildingTypeChoice.setInsertPolicy(QtGui.QComboBox.InsertAtBottom)
self.buildingTypeChoice.setObjectName("buildingTypeChoice")
self.buildingTypeChoice.addItem("")
self.buildingTypeChoice.addItem("")
self.buildingTypeChoice.addItem("")
self.layoutBuildingType.addWidget(self.buildingTypeChoice)
spacerItem2 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutBuildingType.addItem(spacerItem2)
self.layoutManualInput.addLayout(self.layoutBuildingType)
self.layoutCalculationChoice = QtGui.QHBoxLayout()
self.layoutCalculationChoice.setObjectName("layoutCalculationChoice")
self.labelCalculationChoice = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCalculationChoice.sizePolicy().hasHeightForWidth())
self.labelCalculationChoice.setSizePolicy(sizePolicy)
self.labelCalculationChoice.setMinimumSize(QtCore.QSize(0, 17))
self.labelCalculationChoice.setMaximumSize(QtCore.QSize(150, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCalculationChoice.setFont(font)
self.labelCalculationChoice.setObjectName("labelCalculationChoice")
self.layoutCalculationChoice.addWidget(self.labelCalculationChoice)
spacerItem3 = QtGui.QSpacerItem(55, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutCalculationChoice.addItem(spacerItem3)
self.calculationTypeChoice = QtGui.QComboBox(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Fixed)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.calculationTypeChoice.sizePolicy().hasHeightForWidth())
self.calculationTypeChoice.setSizePolicy(sizePolicy)
self.calculationTypeChoice.setMinimumSize(QtCore.QSize(150, 20))
self.calculationTypeChoice.setMaximumSize(QtCore.QSize(150, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.calculationTypeChoice.setFont(font)
self.calculationTypeChoice.setAutoFillBackground(False)
self.calculationTypeChoice.setInsertPolicy(QtGui.QComboBox.InsertAtBottom)
self.calculationTypeChoice.setObjectName("calculationTypeChoice")
self.calculationTypeChoice.addItem("")
self.calculationTypeChoice.addItem("")
self.calculationTypeChoice.addItem("")
self.layoutCalculationChoice.addWidget(self.calculationTypeChoice)
spacerItem4 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutCalculationChoice.addItem(spacerItem4)
self.layoutManualInput.addLayout(self.layoutCalculationChoice)
self.layoutBuildingSystem = QtGui.QHBoxLayout()
self.layoutBuildingSystem.setObjectName("layoutBuildingSystem")

```

```

self.labelbuildingSystem = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelbuildingSystem.sizePolicy().hasHeightForWidth())
self.labelbuildingSystem.setSizePolicy(sizePolicy)
self.labelbuildingSystem.setMinimumSize(QtCore.QSize(0, 17))
self.labelbuildingSystem.setMaximumSize(QtCore.QSize(150, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelbuildingSystem.setFont(font)
self.labelbuildingSystem.setObjectName("labelbuildingSystem")
self.layoutBuildingSystem.addWidget(self.labelbuildingSystem)
spacerItem5 = QtGui.QSpacerItem(55, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutBuildingSystem.addItem(spacerItem5)
self.buildingSystem = QtGui.QComboBox(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Fixed)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.buildingSystem.sizePolicy().hasHeightForWidth())
self.buildingSystem.setSizePolicy(sizePolicy)
self.buildingSystem.setMinimumSize(QtCore.QSize(150, 20))
self.buildingSystem.setMaximumSize(QtCore.QSize(150, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.buildingSystem.setFont(font)
self.buildingSystem.setAutoFillBackground(False)
self.buildingSystem.setInsertPolicy(QtGui.QComboBox.InsertAtBottom)
self.buildingSystem.setObjectName("buildingSystem")
self.buildingSystem.addItem("")
self.buildingSystem.addItem("")
self.buildingSystem.addItem("")
self.layoutBuildingSystem.addWidget(self.buildingSystem)
spacerItem6 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutBuildingSystem.addItem(spacerItem6)
self.layoutManualInput.addLayout(self.layoutBuildingSystem)
spacerItem7 = QtGui.QSpacerItem(50, 20, QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)
self.layoutManualInput.addItem(spacerItem7)
self.layoutIFCFile = QtGui.QHBoxLayout()
self.layoutIFCFile.setObjectName("layoutIFCFile")
self.labelIFCFile = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelIFCFile.sizePolicy().hasHeightForWidth())
self.labelIFCFile.setSizePolicy(sizePolicy)
self.labelIFCFile.setMinimumSize(QtCore.QSize(0, 17))
self.labelIFCFile.setMaximumSize(QtCore.QSize(50, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelIFCFile.setFont(font)
self.labelIFCFile.setObjectName("labelIFCFile")
self.layoutIFCFile.addWidget(self.labelIFCFile)
spacerItem8 = QtGui.QSpacerItem(70, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutIFCFile.addItem(spacerItem8)
self.labelIFCFilePath = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelIFCFilePath.sizePolicy().hasHeightForWidth())
self.labelIFCFilePath.setSizePolicy(sizePolicy)
self.labelIFCFilePath.setMinimumSize(QtCore.QSize(200, 0))
self.labelIFCFilePath.setMaximumSize(QtCore.QSize(200, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setItalic(False)

```

```

font.setWeight(75)
self.labelIFCFilePath.setFont(font)
self.labelIFCFilePath.setText("")
self.labelIFCFilePath.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)
self.labelIFCFilePath.setObjectName("labelIFCFilePath")
self.layoutIFCFile.addWidget(self.labelIFCFilePath)
self.btn_OpenIFC = QtGui.QPushButton(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Fixed)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.btn_OpenIFC.sizePolicy().hasHeightForWidth())
self.btn_OpenIFC.setSizePolicy(sizePolicy)
self.btn_OpenIFC.setMinimumSize(QtCore.QSize(20, 0))
self.btn_OpenIFC.setMaximumSize(QtCore.QSize(20, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.btn_OpenIFC.setFont(font)
self.btn_OpenIFC.setLayoutDirection(QtCore.Qt.LeftToRight)
self.btn_OpenIFC.setObjectName("btn_OpenIFC")
self.layoutIFCFile.addWidget(self.btn_OpenIFC)
spacerItem9 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutIFCFile.addItem(spacerItem9)
self.layoutManualInput.addLayout(self.layoutIFCFile)
self.layoutGFA = QtGui.QHBoxLayout()
self.layoutGFA.setObjectName("layoutGFA")
self.labelGFA = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelGFA.sizePolicy().hasHeightForWidth())
self.labelGFA.setSizePolicy(sizePolicy)
self.labelGFA.setMinimumSize(QtCore.QSize(150, 0))
self.labelGFA.setMaximumSize(QtCore.QSize(150, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelGFA.setFont(font)
self.labelGFA.setObjectName("labelGFA")
self.layoutGFA.addWidget(self.labelGFA)
spacerItem10 = QtGui.QSpacerItem(105, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutGFA.addItem(spacerItem10)
self.GFAInput = QtGui.QLineEdit(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Fixed)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.GFAInput.sizePolicy().hasHeightForWidth())
self.GFAInput.setSizePolicy(sizePolicy)
self.GFAInput.setMinimumSize(QtCore.QSize(100, 20))
self.GFAInput.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.GFAInput.setFont(font)
self.GFAInput.setLayoutDirection(QtCore.Qt.LeftToRight)
self.GFAInput.setText("")
self.GFAInput.setAlignment(QtCore.Qt.AlignCenter)
self.GFAInput.setObjectName("GFAInput")
self.layoutGFA.addWidget(self.GFAInput)
self.labelGFA_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelGFA_unit.sizePolicy().hasHeightForWidth())
self.labelGFA_unit.setSizePolicy(sizePolicy)
self.labelGFA_unit.setMinimumSize(QtCore.QSize(15, 20))
self.labelGFA_unit.setMaximumSize(QtCore.QSize(20, 20))
font = QtGui.QFont()
font.setFamily("Calibri")

```

```

font.setPointSize(11)
self.labelGFA_unit.setFont(font)
self.labelGFA_unit.setLayoutDirection(QtCore.Qt.LeftToRight)
self.labelGFA_unit.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)
self.labelGFA_unit.setObjectName("labelGFA_unit")
self.layoutGFA.addWidget(self.labelGFA_unit)
self.layoutManualInput.addLayout(self.layoutGFA)
self.layoutCalculate = QtGui.QHBoxLayout()
self.layoutCalculate.setObjectName("layoutCalculate")
spacerItem11 = QtGui.QSpacerItem(255, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutCalculate.addItem(spacerItem11)
self.btn_Calculate = QtGui.QPushButton(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.btn_Calculate.sizePolicy().hasHeightForWidth())
self.btn_Calculate.setSizePolicy(sizePolicy)
self.btn_Calculate.setMinimumSize(QtCore.QSize(100, 0))
self.btn_Calculate.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.btn_Calculate.setFont(font)
self.btn_Calculate.setLayoutDirection(QtCore.Qt.LeftToRight)
self.btn_Calculate.setObjectName("btn_Calculate")
self.layoutCalculate.addWidget(self.btn_Calculate)
spacerItem12 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutCalculate.addItem(spacerItem12)
self.layoutManualInput.addLayout(self.layoutCalculate)
self.line_6 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_6.setFont(font)
self.line_6.setFrameShadow(QtGui.QFrame.Sunken)
self.line_6.setFrameShape(QtGui.QFrame.HLine)
self.line_6.setObjectName("line_6")
self.layoutManualInput.addWidget(self.line_6)
spacerItem13 = QtGui.QSpacerItem(50, 20, QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)
self.layoutManualInput.addItem(spacerItem13)
self.layoutResults = QtGui.QVBoxLayout()
self.layoutResults.setObjectName("layoutResults")
self.layoutEI = QtGui.QVBoxLayout()
self.layoutEI.setSpacing(0)
self.layoutEI.setObjectName("layoutEI")
self.labelOverallResults = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelOverallResults.sizePolicy().hasHeightForWidth())
self.labelOverallResults.setSizePolicy(sizePolicy)
self.labelOverallResults.setMinimumSize(QtCore.QSize(360, 20))
self.labelOverallResults.setMaximumSize(QtCore.QSize(16777215, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelOverallResults.setFont(font)
self.labelOverallResults.setFrameShape(QtGui.QFrame.NoFrame)
self.labelOverallResults.setLineWidth(0)
self.labelOverallResults.setScaledContents(False)
self.labelOverallResults.setAlignment(QtCore.Qt.AlignCenter)
self.labelOverallResults.setObjectName("labelOverallResults")
self.layoutEI.addWidget(self.labelOverallResults)
self.line_23 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)

```

```

self.line_23.setFont(font)
self.line_23.setFrameShadow(QtGui.QFrame.Plain)
self.line_23.setFrameShape(QtGui.QFrame.HLine)
self.line_23.setObjectName("line_23")
self.layoutEl.addWidget(self.line_23)
self.layoutEl_Title = QtGui.QHBoxLayout()
self.layoutEl_Title.setSpacing(0)
self.layoutEl_Title.setObjectName("layoutEl_Title")
self.labelEl_category = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEl_category.sizePolicy().hasHeightForWidth())
self.labelEl_category.setSizePolicy(sizePolicy)
self.labelEl_category.setMinimumSize(QtCore.QSize(80, 20))
self.labelEl_category.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelEl_category.setFont(font)
self.labelEl_category.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelEl_category.setFrameShadow(QtGui.QFrame.Plain)
self.labelEl_category.setObjectName("labelEl_category")
self.layoutEl_Title.addWidget(self.labelEl_category)
self.labelEl_impact = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEl_impact.sizePolicy().hasHeightForWidth())
self.labelEl_impact.setSizePolicy(sizePolicy)
self.labelEl_impact.setMinimumSize(QtCore.QSize(100, 20))
self.labelEl_impact.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelEl_impact.setFont(font)
self.labelEl_impact.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelEl_impact.setFrameShape(QtGui.QFrame.NoFrame)
self.labelEl_impact.setLineWidth(0)
self.labelEl_impact.setAlignment(QtCore.Qt.AlignCenter)
self.labelEl_impact.setObjectName("labelEl_impact")
self.layoutEl_Title.addWidget(self.labelEl_impact)
self.labelEl_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEl_unit.sizePolicy().hasHeightForWidth())
self.labelEl_unit.setSizePolicy(sizePolicy)
self.labelEl_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelEl_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelEl_unit.setFont(font)
self.labelEl_unit.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelEl_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelEl_unit.setObjectName("labelEl_unit")
self.layoutEl_Title.addWidget(self.labelEl_unit)
self.labelEl_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEl_SC.sizePolicy().hasHeightForWidth())
self.labelEl_SC.setSizePolicy(sizePolicy)
self.labelEl_SC.setMinimumSize(QtCore.QSize(100, 20))

```



```

self.labelEI_SC.setMaximumSize(QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelEI_SC.setFont(font)
self.labelEI_SC.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelEI_SC.setAlignment(Qt.AlignCenter)
self.labelEI_SC.setObjectName("labelEI_SC")
self.layoutEI_Title.addWidget(self.labelEI_SC)
self.layoutEI.addLayout(self.layoutEI_Title)
self.line_7 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_7.setFont(font)
self.line_7.setFrameShadow(QtGui.QFrame.Plain)
self.line_7.setFrameShape(QtGui.QFrame.HLine)
self.line_7.setObjectName("line_7")
self.layoutEI.addWidget(self.line_7)
self.layoutGWP = QtGui.QHBoxLayout()
self.layoutGWP.setSpacing(0)
self.layoutGWP.setObjectName("layoutGWP")
self.labelGWP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelGWP_name.sizePolicy().hasHeightForWidth())
self.labelGWP_name.setSizePolicy(sizePolicy)
self.labelGWP_name.setMinimumSize(QSize(80, 20))
self.labelGWP_name.setMaximumSize(QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelGWP_name.setFont(font)
self.labelGWP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelGWP_name.setObjectName("labelGWP_name")
self.layoutGWP.addWidget(self.labelGWP_name)
self.labelGWP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelGWP.sizePolicy().hasHeightForWidth())
self.labelGWP.setSizePolicy(sizePolicy)
self.labelGWP.setMinimumSize(QSize(100, 20))
self.labelGWP.setMaximumSize(QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelGWP.setFont(font)
self.labelGWP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelGWP.setText("")
self.labelGWP.setAlignment(Qt.AlignCenter)
self.labelGWP.setObjectName("labelGWP")
self.layoutGWP.addWidget(self.labelGWP)
self.labelGWP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelGWP_unit.sizePolicy().hasHeightForWidth())
self.labelGWP_unit.setSizePolicy(sizePolicy)
self.labelGWP_unit.setMinimumSize(QSize(100, 20))
self.labelGWP_unit.setMaximumSize(QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelGWP_unit.setFont(font)
self.labelGWP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelGWP_unit.setAlignment(Qt.AlignCenter)
self.labelGWP_unit.setObjectName("labelGWP_unit")

```

```

self.layoutGWP.addWidget(self.labelGWP_unit)
self.labelGWP_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelGWP_SC.sizePolicy().hasHeightForWidth())
self.labelGWP_SC.setSizePolicy(sizePolicy)
self.labelGWP_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelGWP_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelGWP_SC.setFont(font)
self.labelGWP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelGWP_SC.setText("")
self.labelGWP_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelGWP_SC.setObjectName("labelGWP_SC")
self.layoutGWP.addWidget(self.labelGWP_SC)
self.layoutEl.addLayout(self.layoutGWP)
self.layoutODP = QtGui.QHBoxLayout()
self.layoutODP.setSpacing(0)
self.layoutODP.setObjectName("layoutODP")
self.labelODP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelODP_name.sizePolicy().hasHeightForWidth())
self.labelODP_name.setSizePolicy(sizePolicy)
self.labelODP_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelODP_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelODP_name.setFont(font)
self.labelODP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelODP_name.setObjectName("labelODP_name")
self.layoutODP.addWidget(self.labelODP_name)
self.labelODP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelODP.sizePolicy().hasHeightForWidth())
self.labelODP.setSizePolicy(sizePolicy)
self.labelODP.setMinimumSize(QtCore.QSize(100, 20))
self.labelODP.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelODP.setFont(font)
self.labelODP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelODP.setText("")
self.labelODP.setAlignment(QtCore.Qt.AlignCenter)
self.labelODP.setObjectName("labelODP")
self.layoutODP.addWidget(self.labelODP)
self.labelODP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelODP_unit.sizePolicy().hasHeightForWidth())
self.labelODP_unit.setSizePolicy(sizePolicy)
self.labelODP_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelODP_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelODP_unit.setFont(font)
self.labelODP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelODP_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelODP_unit.setObjectName("labelODP_unit")
self.layoutODP.addWidget(self.labelODP_unit)
self.labelODP_SC = QtGui.QLabel(self.mainapp)

```

```

sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelODP_SC.sizePolicy().hasHeightForWidth())
self.labelODP_SC.setSizePolicy(sizePolicy)
self.labelODP_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelODP_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelODP_SC.setFont(font)
self.labelODP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelODP_SC.setText("")
self.labelODP_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelODP_SC.setObjectName("labelODP_SC")
self.layoutODP.addWidget(self.labelODP_SC)
self.layoutEl.addLayout(self.layoutODP)
self.layoutPOCP = QtGui.QHBoxLayout()
self.layoutPOCP.setSpacing(0)
self.layoutPOCP.setObjectName("layoutPOCP")
self.labelPOCP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelPOCP_name.sizePolicy().hasHeightForWidth())
self.labelPOCP_name.setSizePolicy(sizePolicy)
self.labelPOCP_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelPOCP_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelPOCP_name.setFont(font)
self.labelPOCP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelPOCP_name.setObjectName("labelPOCP_name")
self.layoutPOCP.addWidget(self.labelPOCP_name)
self.labelPOCP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelPOCP.sizePolicy().hasHeightForWidth())
self.labelPOCP.setSizePolicy(sizePolicy)
self.labelPOCP.setMinimumSize(QtCore.QSize(100, 20))
self.labelPOCP.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelPOCP.setFont(font)
self.labelPOCP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelPOCP.setText("")
self.labelPOCP.setAlignment(QtCore.Qt.AlignCenter)
self.labelPOCP.setObjectName("labelPOCP")
self.layoutPOCP.addWidget(self.labelPOCP)
self.labelPOCP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelPOCP_unit.sizePolicy().hasHeightForWidth())
self.labelPOCP_unit.setSizePolicy(sizePolicy)
self.labelPOCP_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelPOCP_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelPOCP_unit.setFont(font)
self.labelPOCP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelPOCP_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelPOCP_unit.setObjectName("labelPOCP_unit")
self.layoutPOCP.addWidget(self.labelPOCP_unit)
self.labelPOCP_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)

```

```

sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelPOCP_SC.sizePolicy().hasHeightForWidth())
self.labelPOCP_SC.setSizePolicy(sizePolicy)
self.labelPOCP_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelPOCP_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelPOCP_SC.setFont(font)
self.labelPOCP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelPOCP_SC.setText("")
self.labelPOCP_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelPOCP_SC.setObjectName("labelPOCP_SC")
self.layoutPOCP.addWidget(self.labelPOCP_SC)
self.layoutEl.addLayout(self.layoutPOCP)
self.layoutAP = QtGui.QHBoxLayout()
self.layoutAP.setSpacing(0)
self.layoutAP.setObjectName("layoutAP")
self.labelAP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelAP_name.sizePolicy().hasHeightForWidth())
self.labelAP_name.setSizePolicy(sizePolicy)
self.labelAP_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelAP_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelAP_name.setFont(font)
self.labelAP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelAP_name.setObjectName("labelAP_name")
self.layoutAP.addWidget(self.labelAP_name)
self.labelAP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelAP.sizePolicy().hasHeightForWidth())
self.labelAP.setSizePolicy(sizePolicy)
self.labelAP.setMinimumSize(QtCore.QSize(100, 20))
self.labelAP.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelAP.setFont(font)
self.labelAP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelAP.setText("")
self.labelAP.setAlignment(QtCore.Qt.AlignCenter)
self.labelAP.setObjectName("labelAP")
self.layoutAP.addWidget(self.labelAP)
self.labelAP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelAP_unit.sizePolicy().hasHeightForWidth())
self.labelAP_unit.setSizePolicy(sizePolicy)
self.labelAP_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelAP_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelAP_unit.setFont(font)
self.labelAP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelAP_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelAP_unit.setObjectName("labelAP_unit")
self.layoutAP.addWidget(self.labelAP_unit)
self.labelAP_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelAP_SC.sizePolicy().hasHeightForWidth())

```

```

self.labelAP_SC.setSizePolicy(sizePolicy)
self.labelAP_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelAP_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelAP_SC.setFont(font)
self.labelAP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelAP_SC.setText("")
self.labelAP_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelAP_SC.setObjectName("labelAP_SC")
self.layoutAP.addWidget(self.labelAP_SC)
self.layoutEl.addLayout(self.layoutAP)
self.layoutEP = QtGui.QHBoxLayout()
self.layoutEP.setSpacing(0)
self.layoutEP.setObjectName("layoutEP")
self.labelEP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEP_name.sizePolicy().hasHeightForWidth())
self.labelEP_name.setSizePolicy(sizePolicy)
self.labelEP_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelEP_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEP_name.setFont(font)
self.labelEP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEP_name.setObjectName("labelEP_name")
self.layoutEP.addWidget(self.labelEP_name)
self.labelEP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEP.sizePolicy().hasHeightForWidth())
self.labelEP.setSizePolicy(sizePolicy)
self.labelEP.setMinimumSize(QtCore.QSize(100, 20))
self.labelEP.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEP.setFont(font)
self.labelEP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEP.setText("")
self.labelEP.setAlignment(QtCore.Qt.AlignCenter)
self.labelEP.setObjectName("labelEP")
self.layoutEP.addWidget(self.labelEP)
self.labelEP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEP_unit.sizePolicy().hasHeightForWidth())
self.labelEP_unit.setSizePolicy(sizePolicy)
self.labelEP_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelEP_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEP_unit.setFont(font)
self.labelEP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEP_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelEP_unit.setObjectName("labelEP_unit")
self.layoutEP.addWidget(self.labelEP_unit)
self.labelEP_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEP_SC.sizePolicy().hasHeightForWidth())
self.labelEP_SC.setSizePolicy(sizePolicy)
self.labelEP_SC.setMinimumSize(QtCore.QSize(100, 20))

```

```

self.labelEP_SC.setMaximumSize(QCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEP_SC.setFont(font)
self.labelEP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEP_SC.setText("")
self.labelEP_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelEP_SC.setObjectName("labelEP_SC")
self.layoutEP.addWidget(self.labelEP_SC)
self.layoutEl.addLayout(self.layoutEP)
self.layoutHTP = QtGui.QHBoxLayout()
self.layoutHTP.setSpacing(0)
self.layoutHTP.setObjectName("layoutHTP")
self.labelHTP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelHTP_name.sizePolicy().hasHeightForWidth())
self.labelHTP_name.setSizePolicy(sizePolicy)
self.labelHTP_name.setMinimumSize(QCore.QSize(80, 20))
self.labelHTP_name.setMaximumSize(QCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelHTP_name.setFont(font)
self.labelHTP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelHTP_name.setObjectName("labelHTP_name")
self.layoutHTP.addWidget(self.labelHTP_name)
self.labelHTP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelHTP.sizePolicy().hasHeightForWidth())
self.labelHTP.setSizePolicy(sizePolicy)
self.labelHTP.setMinimumSize(QCore.QSize(100, 20))
self.labelHTP.setMaximumSize(QCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelHTP.setFont(font)
self.labelHTP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelHTP.setText("")
self.labelHTP.setAlignment(QtCore.Qt.AlignCenter)
self.labelHTP.setObjectName("labelHTP")
self.layoutHTP.addWidget(self.labelHTP)
self.labelHTP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelHTP_unit.sizePolicy().hasHeightForWidth())
self.labelHTP_unit.setSizePolicy(sizePolicy)
self.labelHTP_unit.setMinimumSize(QCore.QSize(100, 20))
self.labelHTP_unit.setMaximumSize(QCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelHTP_unit.setFont(font)
self.labelHTP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelHTP_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelHTP_unit.setObjectName("labelHTP_unit")
self.layoutHTP.addWidget(self.labelHTP_unit)
self.labelHTP_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelHTP_SC.sizePolicy().hasHeightForWidth())
self.labelHTP_SC.setSizePolicy(sizePolicy)
self.labelHTP_SC.setMinimumSize(QCore.QSize(100, 20))
self.labelHTP_SC.setMaximumSize(QCore.QSize(100, 16777215))
font = QtGui.QFont()

```



```

font.setFamily("Calibri")
font.setPointSize(11)
self.labelHTP_SC.setFont(font)
self.labelHTP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelHTP_SC.setText("")
self.labelHTP_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelHTP_SC.setObjectName("labelHTP_SC")
self.layoutHTP.addWidget(self.labelHTP_SC)
self.layoutEl.addLayout(self.layoutHTP)
self.layoutTETP = QtGui.QHBoxLayout()
self.layoutTETP.setSpacing(0)
self.layoutTETP.setObjectName("layoutTETP")
self.labelTETP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTETP_name.sizePolicy().hasHeightForWidth())
self.labelTETP_name.setSizePolicy(sizePolicy)
self.labelTETP_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelTETP_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelTETP_name.setFont(font)
self.labelTETP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTETP_name.setObjectName("labelTETP_name")
self.layoutTETP.addWidget(self.labelTETP_name)
self.labelTETP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTETP.sizePolicy().hasHeightForWidth())
self.labelTETP.setSizePolicy(sizePolicy)
self.labelTETP.setMinimumSize(QtCore.QSize(100, 20))
self.labelTETP.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelTETP.setFont(font)
self.labelTETP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTETP.setText("")
self.labelTETP.setAlignment(QtCore.Qt.AlignCenter)
self.labelTETP.setObjectName("labelTETP")
self.layoutTETP.addWidget(self.labelTETP)
self.labelTETP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTETP_unit.sizePolicy().hasHeightForWidth())
self.labelTETP_unit.setSizePolicy(sizePolicy)
self.labelTETP_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelTETP_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelTETP_unit.setFont(font)
self.labelTETP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTETP_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelTETP_unit.setObjectName("labelTETP_unit")
self.layoutTETP.addWidget(self.labelTETP_unit)
self.labelTETP_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTETP_SC.sizePolicy().hasHeightForWidth())
self.labelTETP_SC.setSizePolicy(sizePolicy)
self.labelTETP_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelTETP_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)

```



```

self.labelTETP_SC.setFont(font)
self.labelTETP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTETP_SC.setText("")
self.labelTETP_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelTETP_SC.setObjectName("labelTETP_SC")
self.layoutTETP.addWidget(self.labelTETP_SC)
self.layoutEl.addLayout(self.layoutTETP)
self.layoutFAETP = QtGui.QHBoxLayout()
self.layoutFAETP.setSpacing(0)
self.layoutFAETP.setObjectName("layoutFAETP")
self.labelFAETP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelFAETP_name.sizePolicy().hasHeightForWidth())
self.labelFAETP_name.setSizePolicy(sizePolicy)
self.labelFAETP_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelFAETP_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelFAETP_name.setFont(font)
self.labelFAETP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelFAETP_name.setObjectName("labelFAETP_name")
self.layoutFAETP.addWidget(self.labelFAETP_name)
self.labelFAETP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelFAETP.sizePolicy().hasHeightForWidth())
self.labelFAETP.setSizePolicy(sizePolicy)
self.labelFAETP.setMinimumSize(QtCore.QSize(100, 20))
self.labelFAETP.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelFAETP.setFont(font)
self.labelFAETP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelFAETP.setText("")
self.labelFAETP.setAlignment(QtCore.Qt.AlignCenter)
self.labelFAETP.setObjectName("labelFAETP")
self.layoutFAETP.addWidget(self.labelFAETP)
self.labelFAETP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelFAETP_unit.sizePolicy().hasHeightForWidth())
self.labelFAETP_unit.setSizePolicy(sizePolicy)
self.labelFAETP_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelFAETP_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelFAETP_unit.setFont(font)
self.labelFAETP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelFAETP_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelFAETP_unit.setObjectName("labelFAETP_unit")
self.layoutFAETP.addWidget(self.labelFAETP_unit)
self.labelFAETP_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelFAETP_SC.sizePolicy().hasHeightForWidth())
self.labelFAETP_SC.setSizePolicy(sizePolicy)
self.labelFAETP_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelFAETP_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelFAETP_SC.setFont(font)
self.labelFAETP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")

```

```

self.labelFAETP_SC.setText("")
self.labelFAETP_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelFAETP_SC.setObjectName("labelFAETP_SC")
self.layoutFAETP.addWidget(self.labelFAETP_SC)
self.layoutEl.addLayout(self.layoutFAETP)
self.layoutMAETP = QtGui.QHBoxLayout()
self.layoutMAETP.setSpacing(0)
self.layoutMAETP.setObjectName("layoutMAETP")
self.labelMAETP_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelMAETP_name.sizePolicy().hasHeightForWidth())
self.labelMAETP_name.setSizePolicy(sizePolicy)
self.labelMAETP_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelMAETP_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelMAETP_name.setFont(font)
self.labelMAETP_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelMAETP_name.setObjectName("labelMAETP_name")
self.layoutMAETP.addWidget(self.labelMAETP_name)
self.labelMAETP = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelMAETP.sizePolicy().hasHeightForWidth())
self.labelMAETP.setSizePolicy(sizePolicy)
self.labelMAETP.setMinimumSize(QtCore.QSize(100, 20))
self.labelMAETP.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelMAETP.setFont(font)
self.labelMAETP.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelMAETP.setText("")
self.labelMAETP.setAlignment(QtCore.Qt.AlignCenter)
self.labelMAETP.setObjectName("labelMAETP")
self.layoutMAETP.addWidget(self.labelMAETP)
self.labelMAETP_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelMAETP_unit.sizePolicy().hasHeightForWidth())
self.labelMAETP_unit.setSizePolicy(sizePolicy)
self.labelMAETP_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelMAETP_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelMAETP_unit.setFont(font)
self.labelMAETP_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelMAETP_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelMAETP_unit.setObjectName("labelMAETP_unit")
self.layoutMAETP.addWidget(self.labelMAETP_unit)
self.labelMAETP_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelMAETP_SC.sizePolicy().hasHeightForWidth())
self.labelMAETP_SC.setSizePolicy(sizePolicy)
self.labelMAETP_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelMAETP_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelMAETP_SC.setFont(font)
self.labelMAETP_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelMAETP_SC.setText("")
self.labelMAETP_SC.setAlignment(QtCore.Qt.AlignCenter)

```

```

self.labelMAETP_SC.setObjectName("labelMAETP_SC")
self.layoutMAETP.addWidget(self.labelMAETP_SC)
self.layoutEl.addLayout(self.layoutMAETP)
self.layoutADPc = QtGui.QHBoxLayout()
self.layoutADPc.setSpacing(0)
self.layoutADPc.setObjectName("layoutADPc")
self.labelADPc_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelADPc_name.sizePolicy().hasHeightForWidth())
self.labelADPc_name.setSizePolicy(sizePolicy)
self.labelADPc_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelADPc_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelADPc_name.setFont(font)
self.labelADPc_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelADPc_name.setObjectName("labelADPc_name")
self.layoutADPc.addWidget(self.labelADPc_name)
self.labelADPc = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelADPc.sizePolicy().hasHeightForWidth())
self.labelADPc.setSizePolicy(sizePolicy)
self.labelADPc.setMinimumSize(QtCore.QSize(100, 20))
self.labelADPc.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelADPc.setFont(font)
self.labelADPc.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelADPc.setText("")
self.labelADPc.setAlignment(QtCore.Qt.AlignCenter)
self.labelADPc.setObjectName("labelADPc")
self.layoutADPc.addWidget(self.labelADPc)
self.labelADPc_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelADPc_unit.sizePolicy().hasHeightForWidth())
self.labelADPc_unit.setSizePolicy(sizePolicy)
self.labelADPc_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelADPc_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelADPc_unit.setFont(font)
self.labelADPc_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelADPc_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelADPc_unit.setObjectName("labelADPc_unit")
self.layoutADPc.addWidget(self.labelADPc_unit)
self.labelADPc_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelADPc_SC.sizePolicy().hasHeightForWidth())
self.labelADPc_SC.setSizePolicy(sizePolicy)
self.labelADPc_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelADPc_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelADPc_SC.setFont(font)
self.labelADPc_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelADPc_SC.setText("")
self.labelADPc_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelADPc_SC.setObjectName("labelADPc_SC")
self.layoutADPc.addWidget(self.labelADPc_SC)

```

```

self.layoutEl.addLayout(self.layoutADPc)
self.layoutADPf = QtGui.QHBoxLayout()
self.layoutADPf.setSpacing(0)
self.layoutADPf.setObjectName("layoutADPf")
self.labelADPf_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelADPf_name.sizePolicy().hasHeightForWidth())
self.labelADPf_name.setSizePolicy(sizePolicy)
self.labelADPf_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelADPf_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelADPf_name.setFont(font)
self.labelADPf_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelADPf_name.setObjectName("labelADPf_name")
self.layoutADPf.addWidget(self.labelADPf_name)
self.labelADPf = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelADPf.sizePolicy().hasHeightForWidth())
self.labelADPf.setSizePolicy(sizePolicy)
self.labelADPf.setMinimumSize(QtCore.QSize(100, 20))
self.labelADPf.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelADPf.setFont(font)
self.labelADPf.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelADPf.setText("")
self.labelADPf.setAlignment(QtCore.Qt.AlignCenter)
self.labelADPf.setObjectName("labelADPf")
self.layoutADPf.addWidget(self.labelADPf)
self.labelADPf_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelADPf_unit.sizePolicy().hasHeightForWidth())
self.labelADPf_unit.setSizePolicy(sizePolicy)
self.labelADPf_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelADPf_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelADPf_unit.setFont(font)
self.labelADPf_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelADPf_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelADPf_unit.setObjectName("labelADPf_unit")
self.layoutADPf.addWidget(self.labelADPf_unit)
self.labelADPf_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelADPf_SC.sizePolicy().hasHeightForWidth())
self.labelADPf_SC.setSizePolicy(sizePolicy)
self.labelADPf_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelADPf_SC.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelADPf_SC.setFont(font)
self.labelADPf_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelADPf_SC.setText("")
self.labelADPf_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelADPf_SC.setObjectName("labelADPf_SC")
self.layoutADPf.addWidget(self.labelADPf_SC)
self.layoutEl.addLayout(self.layoutADPf)
self.line_8 = QtGui.QFrame(self.mainapp)

```

```

font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_8.setFont(font)
self.line_8.setFrameShadow(QtGui.QFrame.Plain)
self.line_8.setFrameShape(QtGui.QFrame.HLine)
self.line_8.setObjectName("line_8")
self.layoutEl.addWidget(self.line_8)
self.layoutTotalEl = QtGui.QHBoxLayout()
self.layoutTotalEl.setSpacing(0)
self.layoutTotalEl.setObjectName("layoutTotalEl")
self.labelTotalEl_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalEl_name.sizePolicy().hasHeightForWidth())
self.labelTotalEl_name.setSizePolicy(sizePolicy)
self.labelTotalEl_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelTotalEl_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelTotalEl_name.setFont(font)
self.labelTotalEl_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalEl_name.setObjectName("labelTotalEl_name")
self.layoutTotalEl.addWidget(self.labelTotalEl_name)
self.labelTotalEl = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalEl.sizePolicy().hasHeightForWidth())
self.labelTotalEl.setSizePolicy(sizePolicy)
self.labelTotalEl.setMinimumSize(QtCore.QSize(100, 20))
self.labelTotalEl.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelTotalEl.setFont(font)
self.labelTotalEl.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalEl.setText("")
self.labelTotalEl.setObjectName("labelTotalEl")
self.layoutTotalEl.addWidget(self.labelTotalEl)
self.labelTotalEl_unit = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalEl_unit.sizePolicy().hasHeightForWidth())
self.labelTotalEl_unit.setSizePolicy(sizePolicy)
self.labelTotalEl_unit.setMinimumSize(QtCore.QSize(100, 20))
self.labelTotalEl_unit.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelTotalEl_unit.setFont(font)
self.labelTotalEl_unit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalEl_unit.setText("")
self.labelTotalEl_unit.setAlignment(QtCore.Qt.AlignCenter)
self.labelTotalEl_unit.setObjectName("labelTotalEl_unit")
self.layoutTotalEl.addWidget(self.labelTotalEl_unit)
self.labelTotalEl_SC = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalEl_SC.sizePolicy().hasHeightForWidth())
self.labelTotalEl_SC.setSizePolicy(sizePolicy)
self.labelTotalEl_SC.setMinimumSize(QtCore.QSize(100, 20))
self.labelTotalEl_SC.setMaximumSize(QtCore.QSize(100, 16777215))

```

```

font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelTotalEI_SC.setFont(font)
self.labelTotalEI_SC.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalEI_SC.setText("")
self.labelTotalEI_SC.setAlignment(QtCore.Qt.AlignCenter)
self.labelTotalEI_SC.setObjectName("labelTotalEI_SC")
self.layoutTotalEI.addWidget(self.labelTotalEI_SC)
self.layoutEI.addLayout(self.layoutTotalEI)
self.line_9 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_9.setFont(font)
self.line_9.setFrameShadow(QtGui.QFrame.Plain)
self.line_9.setFrameShape(QtGui.QFrame.HLine)
self.line_9.setObjectName("line_9")
self.layoutEI.addWidget(self.line_9)
self.layoutBuildingDecree = QtGui.QHBoxLayout()
self.layoutBuildingDecree.setSpacing(0)
self.layoutBuildingDecree.setObjectName("layoutBuildingDecree")
self.labelBC2012_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelBC2012_name.sizePolicy().hasHeightForWidth())
self.labelBC2012_name.setSizePolicy(sizePolicy)
self.labelBC2012_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelBC2012_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelBC2012_name.setFont(font)
self.labelBC2012_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelBC2012_name.setObjectName("labelBC2012_name")
self.layoutBuildingDecree.addWidget(self.labelBC2012_name)
self.labelBC2012_blank = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelBC2012_blank.sizePolicy().hasHeightForWidth())
self.labelBC2012_blank.setSizePolicy(sizePolicy)
self.labelBC2012_blank.setMinimumSize(QtCore.QSize(50, 20))
self.labelBC2012_blank.setMaximumSize(QtCore.QSize(50, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelBC2012_blank.setFont(font)
self.labelBC2012_blank.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelBC2012_blank.setText("")
self.labelBC2012_blank.setObjectName("labelBC2012_blank")
self.layoutBuildingDecree.addWidget(self.labelBC2012_blank)
self.labelBC2012_ApprovedDeclined = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelBC2012_ApprovedDeclined.sizePolicy().hasHeightForWidth())
self.labelBC2012_ApprovedDeclined.setSizePolicy(sizePolicy)
self.labelBC2012_ApprovedDeclined.setMinimumSize(QtCore.QSize(100, 20))
self.labelBC2012_ApprovedDeclined.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelBC2012_ApprovedDeclined.setFont(font)

```



```

self.labelBC2012_ApprovedDeclined.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelBC2012_ApprovedDeclined.setText("")
self.labelBC2012_ApprovedDeclined.setAlignment(QtCore.Qt.AlignCenter)
self.labelBC2012_ApprovedDeclined.setObjectName("labelBC2012_ApprovedDeclined")
self.layoutBuildingDecree.addWidget(self.labelBC2012_ApprovedDeclined)
self.labelBC2012_SCFinal = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelBC2012_SCFinal.sizePolicy().hasHeightForWidth())
self.labelBC2012_SCFinal.setSizePolicy(sizePolicy)
self.labelBC2012_SCFinal.setMinimumSize(QtCore.QSize(100, 20))
self.labelBC2012_SCFinal.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelBC2012_SCFinal.setFont(font)
self.labelBC2012_SCFinal.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelBC2012_SCFinal.setLineWidth(-7)
self.labelBC2012_SCFinal.setText("")
self.labelBC2012_SCFinal.setAlignment(QtCore.Qt.AlignCenter)
self.labelBC2012_SCFinal.setObjectName("labelBC2012_SCFinal")
self.layoutBuildingDecree.addWidget(self.labelBC2012_SCFinal)
self.layoutEl.addLayout(self.layoutBuildingDecree)
self.line_10 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_10.setFont(font)
self.line_10.setFrameShadow(QtGui.QFrame.Plain)
self.line_10.setLineWidth(1)
self.line_10.setFrameShape(QtGui.QFrame.HLine)
self.line_10.setObjectName("line_10")
self.layoutEl.addWidget(self.line_10)
self.layoutResults.addLayout(self.layoutEl)
self.layoutCost = QtGui.QVBoxLayout()
self.layoutCost.setSpacing(0)
self.layoutCost.setObjectName("layoutCost")
self.line_1 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_1.setFont(font)
self.line_1.setFrameShadow(QtGui.QFrame.Plain)
self.line_1.setLineWidth(1)
self.line_1.setMidLineWidth(5)
self.line_1.setFrameShape(QtGui.QFrame.HLine)
self.line_1.setObjectName("line_1")
self.layoutCost.addWidget(self.line_1)
self.layoutCostClass = QtGui.QHBoxLayout()
self.layoutCostClass.setSpacing(0)
self.layoutCostClass.setObjectName("layoutCostClass")
self.labelCost_empty = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCost_empty.sizePolicy().hasHeightForWidth())
self.labelCost_empty.setSizePolicy(sizePolicy)
self.labelCost_empty.setMinimumSize(QtCore.QSize(80, 20))
self.labelCost_empty.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelCost_empty.setFont(font)
self.labelCost_empty.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelCost_empty.setFrameShadow(QtGui.QFrame.Plain)
self.labelCost_empty.setText("")

```



```

self.labelCost_empty.setObjectName("labelCost_empty")
self.layoutCostClass.addWidget(self.labelCost_empty)
self.line_13 = QtGui.QFrame(self.mainapp)
self.line_13.setFrameShadow(QtGui.QFrame.Plain)
self.line_13.setFrameShape(QtGui.QFrame.VLine)
self.line_13.setObjectName("line_13")
self.layoutCostClass.addWidget(self.line_13)
self.labelClassified = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelClassified.sizePolicy().hasHeightForWidth())
self.labelClassified.setSizePolicy(sizePolicy)
self.labelClassified.setMinimumSize(QtCore.QSize(150, 20))
self.labelClassified.setMaximumSize(QtCore.QSize(150, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelClassified.setFont(font)
self.labelClassified.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelClassified.setAlignment(QtCore.Qt.AlignCenter)
self.labelClassified.setObjectName("labelClassified")
self.layoutCostClass.addWidget(self.labelClassified)
self.line_14 = QtGui.QFrame(self.mainapp)
self.line_14.setFrameShadow(QtGui.QFrame.Plain)
self.line_14.setFrameShape(QtGui.QFrame.VLine)
self.line_14.setObjectName("line_14")
self.layoutCostClass.addWidget(self.line_14)
self.labelSpecif = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelSpecif.sizePolicy().hasHeightForWidth())
self.labelSpecif.setSizePolicy(sizePolicy)
self.labelSpecif.setMinimumSize(QtCore.QSize(150, 20))
self.labelSpecif.setMaximumSize(QtCore.QSize(150, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelSpecif.setFont(font)
self.labelSpecif.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelSpecif.setAlignment(QtCore.Qt.AlignCenter)
self.labelSpecif.setObjectName("labelSpecif")
self.layoutCostClass.addWidget(self.labelSpecif)
self.layoutCost.addLayout(self.layoutCostClass)
self.layoutCostTitle = QtGui.QHBoxLayout()
self.layoutCostTitle.setSpacing(0)
self.layoutCostTitle.setObjectName("layoutCostTitle")
self.labelCost_category = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCost_category.sizePolicy().hasHeightForWidth())
self.labelCost_category.setSizePolicy(sizePolicy)
self.labelCost_category.setMinimumSize(QtCore.QSize(80, 20))
self.labelCost_category.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelCost_category.setFont(font)
self.labelCost_category.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelCost_category.setFrameShadow(QtGui.QFrame.Plain)
self.labelCost_category.setObjectName("labelCost_category")
self.layoutCostTitle.addWidget(self.labelCost_category)
self.line_11 = QtGui.QFrame(self.mainapp)

```

```

self.line_11.setFrameShadow(QtGui.QFrame.Plain)
self.line_11.setFrameShape(QtGui.QFrame.VLine)
self.line_11.setObjectName("line_11")
self.layoutCostTitle.addWidget(self.line_11)
self.labelCost_impact = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCost_impact.sizePolicy().hasHeightForWidth())
self.labelCost_impact.setSizePolicy(sizePolicy)
self.labelCost_impact.setMinimumSize(QtCore.QSize(75, 20))
self.labelCost_impact.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelCost_impact.setFont(font)
self.labelCost_impact.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelCost_impact.setAlignment(QtCore.Qt.AlignCenter)
self.labelCost_impact.setObjectName("labelCost_impact")
self.layoutCostTitle.addWidget(self.labelCost_impact)
self.labelSurface_Classified = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelSurface_Classified.sizePolicy().hasHeightForWidth())
self.labelSurface_Classified.setSizePolicy(sizePolicy)
self.labelSurface_Classified.setMinimumSize(QtCore.QSize(75, 20))
self.labelSurface_Classified.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelSurface_Classified.setFont(font)
self.labelSurface_Classified.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelSurface_Classified.setAlignment(QtCore.Qt.AlignCenter)
self.labelSurface_Classified.setObjectName("labelSurface_Classified")
self.layoutCostTitle.addWidget(self.labelSurface_Classified)
self.line_12 = QtGui.QFrame(self.mainapp)
self.line_12.setFrameShadow(QtGui.QFrame.Plain)
self.line_12.setFrameShape(QtGui.QFrame.VLine)
self.line_12.setObjectName("line_12")
self.layoutCostTitle.addWidget(self.line_12)
self.labelCost_impact_2 = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCost_impact_2.sizePolicy().hasHeightForWidth())
self.labelCost_impact_2.setSizePolicy(sizePolicy)
self.labelCost_impact_2.setMinimumSize(QtCore.QSize(75, 20))
self.labelCost_impact_2.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelCost_impact_2.setFont(font)
self.labelCost_impact_2.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelCost_impact_2.setAlignment(QtCore.Qt.AlignCenter)
self.labelCost_impact_2.setObjectName("labelCost_impact_2")
self.layoutCostTitle.addWidget(self.labelCost_impact_2)
self.labelSurfaceSpecific = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelSurfaceSpecific.sizePolicy().hasHeightForWidth())
self.labelSurfaceSpecific.setSizePolicy(sizePolicy)
self.labelSurfaceSpecific.setMinimumSize(QtCore.QSize(75, 20))
self.labelSurfaceSpecific.setMaximumSize(QtCore.QSize(75, 16777215))

```

```

font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelSurfaceSpecific.setFont(font)
self.labelSurfaceSpecific.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelSurfaceSpecific.setAlignment(QtCore.Qt.AlignCenter)
self.labelSurfaceSpecific.setObjectName("labelSurfaceSpecific")
self.layoutCostTitle.addWidget(self.labelSurfaceSpecific)
self.layoutCost.addLayout(self.layoutCostTitle)
self.line_2 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_2.setFont(font)
self.line_2.setFrameShadow(QtGui.QFrame.Plain)
self.line_2.setLineWidth(1)
self.line_2.setMidLineWidth(5)
self.line_2.setFrameShape(QtGui.QFrame.HLine)
self.line_2.setObjectName("line_2")
self.layoutCost.addWidget(self.line_2)
self.layoutCostInteriorWall = QtGui.QHBoxLayout()
self.layoutCostInteriorWall.setSpacing(0)
self.layoutCostInteriorWall.setObjectName("layoutCostInteriorWall")
self.labelCostInteriorWall_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostInteriorWall_name.sizePolicy().hasHeightForWidth())
self.labelCostInteriorWall_name.setSizePolicy(sizePolicy)
self.labelCostInteriorWall_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelCostInteriorWall_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostInteriorWall_name.setFont(font)
self.labelCostInteriorWall_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostInteriorWall_name.setObjectName("labelCostInteriorWall_name")
self.layoutCostInteriorWall.addWidget(self.labelCostInteriorWall_name)
self.labelCostInteriorWallGeneric = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostInteriorWallGeneric.sizePolicy().hasHeightForWidth())
self.labelCostInteriorWallGeneric.setSizePolicy(sizePolicy)
self.labelCostInteriorWallGeneric.setMinimumSize(QtCore.QSize(75, 20))
self.labelCostInteriorWallGeneric.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostInteriorWallGeneric.setFont(font)
self.labelCostInteriorWallGeneric.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostInteriorWallGeneric.setText("")
self.labelCostInteriorWallGeneric.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)
self.labelCostInteriorWallGeneric.setObjectName("labelCostInteriorWallGeneric")
self.layoutCostInteriorWall.addWidget(self.labelCostInteriorWallGeneric)
self.labelInteriorWallSurfaceGeneric = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelInteriorWallSurfaceGeneric.sizePolicy().hasHeightForWidth())
self.labelInteriorWallSurfaceGeneric.setSizePolicy(sizePolicy)
self.labelInteriorWallSurfaceGeneric.setMinimumSize(QtCore.QSize(75, 20))
self.labelInteriorWallSurfaceGeneric.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelInteriorWallSurfaceGeneric.setFont(font)
self.labelInteriorWallSurfaceGeneric.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelInteriorWallSurfaceGeneric.setText("")

```

```

self.labelInteriorWallSurfaceGeneric.setAlignment(QtCore.Qt.AlignCenter)
self.labelInteriorWallSurfaceGeneric.setObjectName("labelInteriorWallSurfaceGeneric")
self.layoutCostInteriorWall.addWidget(self.labelInteriorWallSurfaceGeneric)
self.labelCostInteriorWallSpecific = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostInteriorWallSpecific.sizePolicy().hasHeightForWidth())
self.labelCostInteriorWallSpecific.setSizePolicy(sizePolicy)
self.labelCostInteriorWallSpecific.setMinimumSize(QtCore.QSize(75, 20))
self.labelCostInteriorWallSpecific.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostInteriorWallSpecific.setFont(font)
self.labelCostInteriorWallSpecific.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostInteriorWallSpecific.setText("")
self.labelCostInteriorWallSpecific.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)
self.labelCostInteriorWallSpecific.setObjectName("labelCostInteriorWallSpecific")
self.layoutCostInteriorWall.addWidget(self.labelCostInteriorWallSpecific)
self.labelInteriorWallSurfaceSpecific = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelInteriorWallSurfaceSpecific.sizePolicy().hasHeightForWidth())
self.labelInteriorWallSurfaceSpecific.setSizePolicy(sizePolicy)
self.labelInteriorWallSurfaceSpecific.setMinimumSize(QtCore.QSize(75, 20))
self.labelInteriorWallSurfaceSpecific.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelInteriorWallSurfaceSpecific.setFont(font)
self.labelInteriorWallSurfaceSpecific.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelInteriorWallSurfaceSpecific.setText("")
self.labelInteriorWallSurfaceSpecific.setAlignment(QtCore.Qt.AlignCenter)
self.labelInteriorWallSurfaceSpecific.setObjectName("labelInteriorWallSurfaceSpecific")
self.layoutCostInteriorWall.addWidget(self.labelInteriorWallSurfaceSpecific)
self.layoutCost.addLayout(self.layoutCostInteriorWall)
self.layoutCostFloor = QtGui.QHBoxLayout()
self.layoutCostFloor.setSpacing(0)
self.layoutCostFloor.setObjectName("layoutCostFloor")
self.labelCostFloor_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostFloor_name.sizePolicy().hasHeightForWidth())
self.labelCostFloor_name.setSizePolicy(sizePolicy)
self.labelCostFloor_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelCostFloor_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostFloor_name.setFont(font)
self.labelCostFloor_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostFloor_name.setObjectName("labelCostFloor_name")
self.layoutCostFloor.addWidget(self.labelCostFloor_name)
self.labelCostFloorGeneric = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostFloorGeneric.sizePolicy().hasHeightForWidth())
self.labelCostFloorGeneric.setSizePolicy(sizePolicy)
self.labelCostFloorGeneric.setMinimumSize(QtCore.QSize(75, 20))
self.labelCostFloorGeneric.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostFloorGeneric.setFont(font)
self.labelCostFloorGeneric.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostFloorGeneric.setText("")
self.labelCostFloorGeneric.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)

```

```

self.labelCostFloorGeneric.setObjectName("labelCostFloorGeneric")
self.layoutCostFloor.addWidget(self.labelCostFloorGeneric)
self.labelFloorSurfaceGeneric = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelFloorSurfaceGeneric.sizePolicy().hasHeightForWidth())
self.labelFloorSurfaceGeneric.setSizePolicy(sizePolicy)
self.labelFloorSurfaceGeneric.setMinimumSize(QtCore.QSize(75, 20))
self.labelFloorSurfaceGeneric.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelFloorSurfaceGeneric.setFont(font)
self.labelFloorSurfaceGeneric.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelFloorSurfaceGeneric.setText("")
self.labelFloorSurfaceGeneric.setAlignment(QtCore.Qt.AlignCenter)
self.labelFloorSurfaceGeneric.setObjectName("labelFloorSurfaceGeneric")
self.layoutCostFloor.addWidget(self.labelFloorSurfaceGeneric)
self.labelCostFloorSpecific = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostFloorSpecific.sizePolicy().hasHeightForWidth())
self.labelCostFloorSpecific.setSizePolicy(sizePolicy)
self.labelCostFloorSpecific.setMinimumSize(QtCore.QSize(75, 20))
self.labelCostFloorSpecific.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostFloorSpecific.setFont(font)
self.labelCostFloorSpecific.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostFloorSpecific.setText("")
self.labelCostFloorSpecific.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)
self.labelCostFloorSpecific.setObjectName("labelCostFloorSpecific")
self.layoutCostFloor.addWidget(self.labelCostFloorSpecific)
self.labelFloorSurfaceSpecific = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelFloorSurfaceSpecific.sizePolicy().hasHeightForWidth())
self.labelFloorSurfaceSpecific.setSizePolicy(sizePolicy)
self.labelFloorSurfaceSpecific.setMinimumSize(QtCore.QSize(75, 20))
self.labelFloorSurfaceSpecific.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelFloorSurfaceSpecific.setFont(font)
self.labelFloorSurfaceSpecific.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelFloorSurfaceSpecific.setText("")
self.labelFloorSurfaceSpecific.setAlignment(QtCore.Qt.AlignCenter)
self.labelFloorSurfaceSpecific.setObjectName("labelFloorSurfaceSpecific")
self.layoutCostFloor.addWidget(self.labelFloorSurfaceSpecific)
self.layoutCost.addLayout(self.layoutCostFloor)
self.layoutCostExteriorWall = QtGui.QHBoxLayout()
self.layoutCostExteriorWall.setSpacing(0)
self.layoutCostExteriorWall.setObjectName("layoutCostExteriorWall")
self.labelCostExteriorWall_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostExteriorWall_name.sizePolicy().hasHeightForWidth())
self.labelCostExteriorWall_name.setSizePolicy(sizePolicy)
self.labelCostExteriorWall_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelCostExteriorWall_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostExteriorWall_name.setFont(font)
self.labelCostExteriorWall_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostExteriorWall_name.setObjectName("labelCostExteriorWall_name")

```

```

self.layoutCostExteriorWall.addWidget(self.labelCostExteriorWall_name)
self.labelCostExteriorWallGeneric = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostExteriorWallGeneric.sizePolicy().hasHeightForWidth())
self.labelCostExteriorWallGeneric.setSizePolicy(sizePolicy)
self.labelCostExteriorWallGeneric.setMinimumSize(QtCore.QSize(75, 20))
self.labelCostExteriorWallGeneric.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostExteriorWallGeneric.setFont(font)
self.labelCostExteriorWallGeneric.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostExteriorWallGeneric.setText("")
self.labelCostExteriorWallGeneric.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)
self.labelCostExteriorWallGeneric.setObjectName("labelCostExteriorWallGeneric")
self.layoutCostExteriorWall.addWidget(self.labelCostExteriorWallGeneric)
self.labelExteriorWallSurfaceGeneric = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelExteriorWallSurfaceGeneric.sizePolicy().hasHeightForWidth())
self.labelExteriorWallSurfaceGeneric.setSizePolicy(sizePolicy)
self.labelExteriorWallSurfaceGeneric.setMinimumSize(QtCore.QSize(75, 20))
self.labelExteriorWallSurfaceGeneric.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelExteriorWallSurfaceGeneric.setFont(font)
self.labelExteriorWallSurfaceGeneric.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelExteriorWallSurfaceGeneric.setText("")
self.labelExteriorWallSurfaceGeneric.setAlignment(QtCore.Qt.AlignCenter)
self.labelExteriorWallSurfaceGeneric.setObjectName("labelExteriorWallSurfaceGeneric")
self.layoutCostExteriorWall.addWidget(self.labelExteriorWallSurfaceGeneric)
self.labelCostExteriorWallSpecific = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelCostExteriorWallSpecific.sizePolicy().hasHeightForWidth())
self.labelCostExteriorWallSpecific.setSizePolicy(sizePolicy)
self.labelCostExteriorWallSpecific.setMinimumSize(QtCore.QSize(75, 20))
self.labelCostExteriorWallSpecific.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelCostExteriorWallSpecific.setFont(font)
self.labelCostExteriorWallSpecific.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelCostExteriorWallSpecific.setText("")
self.labelCostExteriorWallSpecific.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)
self.labelCostExteriorWallSpecific.setObjectName("labelCostExteriorWallSpecific")
self.layoutCostExteriorWall.addWidget(self.labelCostExteriorWallSpecific)
self.labelExteriorWallSurfaceSpecific = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelExteriorWallSurfaceSpecific.sizePolicy().hasHeightForWidth())
self.labelExteriorWallSurfaceSpecific.setSizePolicy(sizePolicy)
self.labelExteriorWallSurfaceSpecific.setMinimumSize(QtCore.QSize(75, 20))
self.labelExteriorWallSurfaceSpecific.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelExteriorWallSurfaceSpecific.setFont(font)
self.labelExteriorWallSurfaceSpecific.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelExteriorWallSurfaceSpecific.setText("")
self.labelExteriorWallSurfaceSpecific.setAlignment(QtCore.Qt.AlignCenter)
self.labelExteriorWallSurfaceSpecific.setObjectName("labelExteriorWallSurfaceSpecific")
self.layoutCostExteriorWall.addWidget(self.labelExteriorWallSurfaceSpecific)
self.layoutCost.addLayout(self.layoutCostExteriorWall)
self.line_3 = QtGui.QFrame(self.mainapp)

```



```

font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_3.setFont(font)
self.line_3.setFrameShadow(QtGui.QFrame.Plain)
self.line_3.setLineWidth(1)
self.line_3.setFrameShape(QtGui.QFrame.HLine)
self.line_3.setObjectName("line_3")
self.layoutCost.addWidget(self.line_3)
self.layoutTotalCost = QtGui.QHBoxLayout()
self.layoutTotalCost.setSpacing(0)
self.layoutTotalCost.setObjectName("layoutTotalCost")
self.labelTotalCost_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalCost_name.sizePolicy().hasHeightForWidth())
self.labelTotalCost_name.setSizePolicy(sizePolicy)
self.labelTotalCost_name.setMinimumSize(QtCore.QSize(80, 20))
self.labelTotalCost_name.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelTotalCost_name.setFont(font)
self.labelTotalCost_name.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalCost_name.setObjectName("labelTotalCost_name")
self.layoutTotalCost.addWidget(self.labelTotalCost_name)
self.labelTotalCostGeneric = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalCostGeneric.sizePolicy().hasHeightForWidth())
self.labelTotalCostGeneric.setSizePolicy(sizePolicy)
self.labelTotalCostGeneric.setMinimumSize(QtCore.QSize(75, 20))
self.labelTotalCostGeneric.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelTotalCostGeneric.setFont(font)
self.labelTotalCostGeneric.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalCostGeneric.setText("")
self.labelTotalCostGeneric.setAlignment(QtCore.Qt.AlignRight|QtCore.Qt.AlignTrailing|QtCore.Qt.AlignVCenter)
self.labelTotalCostGeneric.setObjectName("labelTotalCostGeneric")
self.layoutTotalCost.addWidget(self.labelTotalCostGeneric)
self.labelTotalCost_empty = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalCost_empty.sizePolicy().hasHeightForWidth())
self.labelTotalCost_empty.setSizePolicy(sizePolicy)
self.labelTotalCost_empty.setMinimumSize(QtCore.QSize(75, 20))
self.labelTotalCost_empty.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelTotalCost_empty.setFont(font)
self.labelTotalCost_empty.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalCost_empty.setText("")
self.labelTotalCost_empty.setAlignment(QtCore.Qt.AlignCenter)
self.labelTotalCost_empty.setObjectName("labelTotalCost_empty")
self.layoutTotalCost.addWidget(self.labelTotalCost_empty)
self.labelTotalCostSpecific = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalCostSpecific.sizePolicy().hasHeightForWidth())
self.labelTotalCostSpecific.setSizePolicy(sizePolicy)

```



```

self.labelTotalCostSpecific.setMinimumSize(QtCore.QSize(75, 20))
self.labelTotalCostSpecific.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelTotalCostSpecific.setFont(font)
self.labelTotalCostSpecific.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalCostSpecific.setText("")
self.labelTotalCostSpecific.setAlignment(QtCore.Qt.AlignRight | QtCore.Qt.AlignTrailing | QtCore.Qt.AlignVCenter)
self.labelTotalCostSpecific.setObjectName("labelTotalCostSpecific")
self.layoutTotalCost.addWidget(self.labelTotalCostSpecific)
self.labelTotalCost_empty_2 = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelTotalCost_empty_2.sizePolicy().hasHeightForWidth())
self.labelTotalCost_empty_2.setSizePolicy(sizePolicy)
self.labelTotalCost_empty_2.setMinimumSize(QtCore.QSize(75, 20))
self.labelTotalCost_empty_2.setMaximumSize(QtCore.QSize(75, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelTotalCost_empty_2.setFont(font)
self.labelTotalCost_empty_2.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelTotalCost_empty_2.setText("")
self.labelTotalCost_empty_2.setAlignment(QtCore.Qt.AlignCenter)
self.labelTotalCost_empty_2.setObjectName("labelTotalCost_empty_2")
self.layoutTotalCost.addWidget(self.labelTotalCost_empty_2)
self.layoutCost.addLayout(self.layoutTotalCost)
self.line_4 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_4.setFont(font)
self.line_4.setFrameShadow(QtGui.QFrame.Plain)
self.line_4.setLineWidth(1)
self.line_4.setMidLineWidth(5)
self.line_4 setFrameShape(QtGui.QFrame.HLine)
self.line_4.setObjectName("line_4")
self.layoutCost.addWidget(self.line_4)
self.layoutResults.addLayout(self.layoutCost)
self.layoutManualInput.addLayout(self.layoutResults)
self.gridLayout.addLayout(self.layoutManualInput, 0, 0, 1, 1)
self.layoutRightBar = QtGui.QVBoxLayout()
self.layoutRightBar.setSizeConstraint(QtGui.QLayout.SetMinimumSize)
self.layoutRightBar.setObjectName("layoutRightBar")
self.tabWidget = QtGui.QTabWidget(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.MinimumExpanding, QtGui.QSizePolicy.Expanding)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.tabWidget.sizePolicy().hasHeightForWidth())
self.tabWidget.setSizePolicy(sizePolicy)
self.tabWidget.setMinimumSize(QtCore.QSize(0, 0))
self.tabWidget.setMaximumSize(QtCore.QSize(16777215, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(13)
self.tabWidget.setFont(font)
self.tabWidget.setContextMenuPolicy(QtCore.Qt.PreventContextMenu)
self.tabWidget.setObjectName("tabWidget")
self.overviewTab = QtGui.QWidget()
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.overviewTab.sizePolicy().hasHeightForWidth())
self.overviewTab.setSizePolicy(sizePolicy)
self.overviewTab.setObjectName("overviewTab")
self.tabWidget.addTab(self.overviewTab, "")
self.layoutRightBar.addWidget(self.tabWidget)

```

```

self.layoutRightBottom = QtGui.QHBoxLayout()
self.layoutRightBottom.setObjectName("layoutRightBottom")
self.layoutOptimalSolution = QtGui.QVBoxLayout()
self.layoutOptimalSolution.setSpacing(0)
self.layoutOptimalSolution.setObjectName("layoutOptimalSolution")
self.labelImportanceOfImpact = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelImportanceOfImpact.sizePolicy().hasHeightForWidth())
self.labelImportanceOfImpact.setSizePolicy(sizePolicy)
self.labelImportanceOfImpact.setMinimumSize(QtCore.QSize(360, 20))
self.labelImportanceOfImpact.setMaximumSize(QtCore.QSize(16777215, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelImportanceOfImpact.setFont(font)
self.labelImportanceOfImpact.setStyleSheet("R")
self.labelImportanceOfImpact.setFrameShape(QtGui.QFrame.NoFrame)
self.labelImportanceOfImpact.setLineWidth(0)
self.labelImportanceOfImpact.setScaledContents(False)
self.labelImportanceOfImpact.setAlignment(QtCore.Qt.AlignCenter)
self.labelImportanceOfImpact.setObjectName("labelImportanceOfImpact")
self.layoutOptimalSolution.addWidget(self.labelImportanceOfImpact)
self.line_21 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_21.setFont(font)
self.line_21.setFrameShadow(QtGui.QFrame.Sunken)
self.line_21.setFrameShape(QtGui.QFrame.HLine)
self.line_21.setObjectName("line_21")
self.layoutOptimalSolution.addWidget(self.line_21)
self.layoutImportanceOfImpact = QtGui.QHBoxLayout()
self.layoutImportanceOfImpact.setObjectName("layoutImportanceOfImpact")
spacerItem14 = QtGui.QSpacerItem(50, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutImportanceOfImpact.addItem(spacerItem14)
self.environmentalImpact = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.environmentalImpact.sizePolicy().hasHeightForWidth())
self.environmentalImpact.setSizePolicy(sizePolicy)
self.environmentalImpact.setMinimumSize(QtCore.QSize(0, 17))
self.environmentalImpact.setMaximumSize(QtCore.QSize(150, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.environmentalImpact.setFont(font)
self.environmentalImpact.setObjectName("environmentalImpact")
self.layoutImportanceOfImpact.addWidget(self.environmentalImpact)
spacerItem15 = QtGui.QSpacerItem(10, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutImportanceOfImpact.addItem(spacerItem15)
self.slider = QtGui.QSlider(self.mainapp)
self.slider.setOrientation(QtCore.Qt.Horizontal)
self.slider.setObjectName("slider")
self.slider.setMinimum(1)
self.slider.setMaximum(9)
self.slider.setValue(5)
self.slider.setTickInterval(1)
self.slider.setTickPosition(QSlider.TicksBelow)
self.layoutImportanceOfImpact.addWidget(self.slider)
spacerItem16 = QtGui.QSpacerItem(10, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutImportanceOfImpact.addItem(spacerItem16)
self.economicImpact = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.economicImpact.sizePolicy().hasHeightForWidth())

```

```

self.economicImpact.setSizePolicy(sizePolicy)
self.economicImpact.setMinimumSize(QtCore.QSize(0, 17))
self.economicImpact.setMaximumSize(QtCore.QSize(150, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.economicImpact.setFont(font)
self.economicImpact.setObjectName("economicImpact")
self.layoutImportanceofImpact.addWidget(self.economicImpact)
spacerItem17 = QtGui.QSpacerItem(50, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutImportanceofImpact.addItem(spacerItem17)
self.layoutOptimalSolution.addLayout(self.layoutImportanceofImpact)
self.layoutGenerate = QtGui.QHBoxLayout()
self.layoutGenerate.setObjectName("layoutGenerate")
spacerItem18 = QtGui.QSpacerItem(220, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutGenerate.addItem(spacerItem18)
spacerItem23 = QtGui.QSpacerItem(0, 0, QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Minimum)
self.layoutRightBottom.addItem(spacerItem23)
self.btn_Generate = QtGui.QPushButton(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.btn_Generate.sizePolicy().hasHeightForWidth())
self.btn_Generate.setSizePolicy(sizePolicy)
self.btn_Generate.setMinimumSize(QtCore.QSize(100, 0))
self.btn_Generate.setMaximumSize(QtCore.QSize(100, 16777215))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.btn_Generate.setFont(font)
self.btn_Generate.setLayoutDirection(QtCore.Qt.LeftToRight)
self.btn_Generate.setObjectName("btn_Generate")
self.layoutGenerate.addWidget(self.btn_Generate)
spacerItem19 = QtGui.QSpacerItem(225, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
self.layoutGenerate.addItem(spacerItem19)
self.layoutOptimalSolution.addLayout(self.layoutGenerate)
spacerItem20 = QtGui.QSpacerItem(50, 5, QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)
self.layoutOptimalSolution.addItem(spacerItem20)
self.line_18 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_18.setFont(font)
self.line_18.setFrameShadow(QtGui.QFrame.Sunken)
self.line_18.setFrameShape(QtGui.QFrame.HLine)
self.line_18.setObjectName("line_18")
self.layoutOptimalSolution.addWidget(self.line_18)
spacerItem21 = QtGui.QSpacerItem(50, 20, QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Fixed)
self.layoutOptimalSolution.addItem(spacerItem21)
self.layoutTitleOptimalSolution = QtGui.QHBoxLayout()
self.layoutTitleOptimalSolution.setObjectName("layoutTitleOptimalSolution")
self.labelEmpty_2 = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Maximum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEmpty_2.sizePolicy().hasHeightForWidth())
self.labelEmpty_2.setSizePolicy(sizePolicy)
self.labelEmpty_2.setMinimumSize(QtCore.QSize(120, 20))
self.labelEmpty_2.setMaximumSize(QtCore.QSize(16777215, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelEmpty_2.setFont(font)
self.labelEmpty_2.setStyleSheet(" R")
self.labelEmpty_2.setText("")
self.labelEmpty_2.setAlignment(QtCore.Qt.AlignCenter)
self.labelEmpty_2.setObjectName("labelEmpty_2")

```

```

self.layoutTitleOptimalSolution.addWidget(self.labelEmpty_2)
self.labelOptimalSolution_name = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelOptimalSolution_name.sizePolicy().hasHeightForWidth())
self.labelOptimalSolution_name.setSizePolicy(sizePolicy)
self.labelOptimalSolution_name.setMinimumSize(QtCore.QSize(360, 20))
self.labelOptimalSolution_name.setMaximumSize(QtCore.QSize(16777215, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelOptimalSolution_name.setFont(font)
self.labelOptimalSolution_name.setStyleSheet(" R")
self.labelOptimalSolution_name setFrameShape(QtGui.QFrame.NoFrame)
self.labelOptimalSolution_name.setLineWidth(0)
self.labelOptimalSolution_name.setScaledContents(False)
self.labelOptimalSolution_name.setAlignment(QtCore.Qt.AlignCenter)
self.labelOptimalSolution_name.setObjectName("labelOptimalSolution_name")
self.layoutTitleOptimalSolution.addWidget(self.labelOptimalSolution_name)
self.layoutOptimalSolution.addLayout(self.layoutTitleOptimalSolution)
self.line_5 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_5.setFont(font)
self.line_5.setFrameShadow(QtGui.QFrame.Plain)
self.line_5.setLineWidth(1)
self.line_5 setFrameShape(QtGui.QFrame.HLine)
self.line_5.setObjectName("line_5")
self.layoutOptimalSolution.addWidget(self.line_5)
self.layoutTitleOptimalSolutionColumns = QtGui.QHBoxLayout()
self.layoutTitleOptimalSolutionColumns.setSpacing(0)
self.layoutTitleOptimalSolutionColumns.setObjectName("layoutTitleOptimalSolutionColumns")
self.labelElements = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Maximum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelElements.sizePolicy().hasHeightForWidth())
self.labelElements.setSizePolicy(sizePolicy)
self.labelElements.setMinimumSize(QtCore.QSize(120, 20))
self.labelElements.setMaximumSize(QtCore.QSize(16777215, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelElements.setFont(font)
self.labelElements.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelElements.setFrameShadow(QtGui.QFrame.Plain)
self.labelElements.setObjectName("labelElements")
self.layoutTitleOptimalSolutionColumns.addWidget(self.labelElements)
self.labelEnvironmental = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEnvironmental.sizePolicy().hasHeightForWidth())
self.labelEnvironmental.setSizePolicy(sizePolicy)
self.labelEnvironmental.setMinimumSize(QtCore.QSize(150, 20))
self.labelEnvironmental.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelEnvironmental.setFont(font)
self.labelEnvironmental.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelEnvironmental setFrameShape(QtGui.QFrame.NoFrame)
self.labelEnvironmental.setLineWidth(0)

```

```

self.labelEnvironmental.setAlignment(QtCore.Qt.AlignCenter)
self.labelEnvironmental.setObjectName("labelEnvironmental")
self.layoutTitleOptimalSolutionColumns.addWidget(self.labelEnvironmental)
self.labelEconomic = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEconomic.sizePolicy().hasHeightForWidth())
self.labelEconomic.setSizePolicy(sizePolicy)
self.labelEconomic.setMinimumSize(QtCore.QSize(150, 20))
self.labelEconomic.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelEconomic.setFont(font)
self.labelEconomic.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelEconomic.setAlignment(QtCore.Qt.AlignCenter)
self.labelEconomic.setObjectName("labelEconomic")
self.layoutTitleOptimalSolutionColumns.addWidget(self.labelEconomic)
self.labelOverallSolution = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Minimum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelOverallSolution.sizePolicy().hasHeightForWidth())
self.labelOverallSolution.setSizePolicy(sizePolicy)
self.labelOverallSolution.setMinimumSize(QtCore.QSize(150, 20))
self.labelOverallSolution.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
font.setBold(True)
font.setWeight(75)
self.labelOverallSolution.setFont(font)
self.labelOverallSolution.setStyleSheet("background-color: rgb(91, 195, 200);")
self.labelOverallSolution.setAlignment(QtCore.Qt.AlignCenter)
self.labelOverallSolution.setObjectName("labelOverallSolution")
self.layoutTitleOptimalSolutionColumns.addWidget(self.labelOverallSolution)
self.layoutOptimalSolution.addLayout(self.layoutTitleOptimalSolutionColumns)
self.line_19 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_19.setFont(font)
self.line_19.setFrameShadow(QtGui.QFrame.Plain)
self.line_19.setFrameShape(QtGui.QFrame.HLine)
self.line_19.setObjectName("line_19")
self.layoutOptimalSolution.addWidget(self.line_19)
self.layoutInteriorWall = QtGui.QHBoxLayout()
self.layoutInteriorWall.setSpacing(0)
self.layoutInteriorWall.setObjectName("layoutInteriorWall")
self.labelName_InteriorWall = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Maximum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelName_InteriorWall.sizePolicy().hasHeightForWidth())
self.labelName_InteriorWall.setSizePolicy(sizePolicy)
self.labelName_InteriorWall.setMinimumSize(QtCore.QSize(120, 20))
self.labelName_InteriorWall.setMaximumSize(QtCore.QSize(16777215, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelName_InteriorWall.setFont(font)
self.labelName_InteriorWall.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelName_InteriorWall.setObjectName("labelName_InteriorWall")
self.layoutInteriorWall.addWidget(self.labelName_InteriorWall)
self.labelEnvironmental_InteriorWall = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)

```

```

sizePolicy.setHeightForWidth(self.labelEnvironmental_InteriorWall.sizePolicy().hasHeightForWidth())
self.labelEnvironmental_InteriorWall.setSizePolicy(sizePolicy)
self.labelEnvironmental_InteriorWall.setMinimumSize(QtCore.QSize(150, 20))
self.labelEnvironmental_InteriorWall.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEnvironmental_InteriorWall.setFont(font)
self.labelEnvironmental_InteriorWall.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEnvironmental_InteriorWall.setText("")
self.labelEnvironmental_InteriorWall.setAlignment(QtCore.Qt.AlignCenter)
self.labelEnvironmental_InteriorWall.setObjectName("labelEnvironmental_InteriorWall")
self.layoutInteriorWall.addWidget(self.labelEnvironmental_InteriorWall)
self.labelEconomic_InteriorWall = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEconomic_InteriorWall.sizePolicy().hasHeightForWidth())
self.labelEconomic_InteriorWall.setSizePolicy(sizePolicy)
self.labelEconomic_InteriorWall.setMinimumSize(QtCore.QSize(150, 20))
self.labelEconomic_InteriorWall.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEconomic_InteriorWall.setFont(font)
self.labelEconomic_InteriorWall.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEconomic_InteriorWall.setAlignment(QtCore.Qt.AlignCenter)
self.labelEconomic_InteriorWall.setObjectName("labelEconomic_InteriorWall")
self.layoutInteriorWall.addWidget(self.labelEconomic_InteriorWall)
self.labelOverall_InteriorWall = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelOverall_InteriorWall.sizePolicy().hasHeightForWidth())
self.labelOverall_InteriorWall.setSizePolicy(sizePolicy)
self.labelOverall_InteriorWall.setMinimumSize(QtCore.QSize(150, 20))
self.labelOverall_InteriorWall.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelOverall_InteriorWall.setFont(font)
self.labelOverall_InteriorWall.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelOverall_InteriorWall.setText("")
self.labelOverall_InteriorWall.setAlignment(QtCore.Qt.AlignCenter)
self.labelOverall_InteriorWall.setObjectName("labelOverall_InteriorWall")
self.layoutInteriorWall.addWidget(self.labelOverall_InteriorWall)
self.layoutOptimalSolution.addLayout(self.layoutInteriorWall)
self.layoutFloor = QtGui.QHBoxLayout()
self.layoutFloor.setSpacing(0)
self.layoutFloor.setObjectName("layoutFloor")
self.labelName_Floor = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Maximum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelName_Floor.sizePolicy().hasHeightForWidth())
self.labelName_Floor.setSizePolicy(sizePolicy)
self.labelName_Floor.setMinimumSize(QtCore.QSize(120, 20))
self.labelName_Floor.setMaximumSize(QtCore.QSize(16777215, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelName_Floor.setFont(font)
self.labelName_Floor.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelName_Floor.setObjectName("labelName_Floor")
self.layoutFloor.addWidget(self.labelName_Floor)
self.labelEnvironmental_Floor = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEnvironmental_Floor.sizePolicy().hasHeightForWidth())
self.labelEnvironmental_Floor.setSizePolicy(sizePolicy)

```



```

self.labelEnvironmental_Floor.setMinimumSize(QtCore.QSize(150, 20))
self.labelEnvironmental_Floor.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEnvironmental_Floor.setFont(font)
self.labelEnvironmental_Floor.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEnvironmental_Floor.setText("")
self.labelEnvironmental_Floor.setAlignment(QtCore.Qt.AlignCenter)
self.labelEnvironmental_Floor.setObjectName("labelEnvironmental_Floor")
self.layoutFloor.addWidget(self.labelEnvironmental_Floor)
self.labelEconomic_Floor = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEconomic_Floor.sizePolicy().hasHeightForWidth())
self.labelEconomic_Floor.setSizePolicy(sizePolicy)
self.labelEconomic_Floor.setMinimumSize(QtCore.QSize(150, 20))
self.labelEconomic_Floor.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEconomic_Floor.setFont(font)
self.labelEconomic_Floor.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEconomic_Floor.setAlignment(QtCore.Qt.AlignCenter)
self.labelEconomic_Floor.setObjectName("labelEconomic_Floor")
self.layoutFloor.addWidget(self.labelEconomic_Floor)
self.labelOverall_Floor = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelOverall_Floor.sizePolicy().hasHeightForWidth())
self.labelOverall_Floor.setSizePolicy(sizePolicy)
self.labelOverall_Floor.setMinimumSize(QtCore.QSize(150, 20))
self.labelOverall_Floor.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelOverall_Floor.setFont(font)
self.labelOverall_Floor.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelOverall_Floor.setText("")
self.labelOverall_Floor.setAlignment(QtCore.Qt.AlignCenter)
self.labelOverall_Floor.setObjectName("labelOverall_Floor")
self.layoutFloor.addWidget(self.labelOverall_Floor)
self.layoutOptimalSolution.addLayout(self.layoutFloor)
self.layoutExteriorWall = QtGui.QHBoxLayout()
self.layoutExteriorWall.setSpacing(0)
self.layoutExteriorWall.setObjectName("layoutExteriorWall")
self.labelName_ExteriorWall = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Maximum, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelName_ExteriorWall.sizePolicy().hasHeightForWidth())
self.labelName_ExteriorWall.setSizePolicy(sizePolicy)
self.labelName_ExteriorWall.setMinimumSize(QtCore.QSize(120, 20))
self.labelName_ExteriorWall.setMaximumSize(QtCore.QSize(16777215, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelName_ExteriorWall.setFont(font)
self.labelName_ExteriorWall.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelName_ExteriorWall.setObjectName("labelName_ExteriorWall")
self.layoutExteriorWall.addWidget(self.labelName_ExteriorWall)
self.labelEnvironmental_ExteriorWall = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEnvironmental_ExteriorWall.sizePolicy().hasHeightForWidth())
self.labelEnvironmental_ExteriorWall.setSizePolicy(sizePolicy)
self.labelEnvironmental_ExteriorWall.setMinimumSize(QtCore.QSize(150, 20))
self.labelEnvironmental_ExteriorWall.setMaximumSize(QtCore.QSize(100, 20))

```



```

font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEnvironmental_ExteriorWall.setFont(font)
self.labelEnvironmental_ExteriorWall.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEnvironmental_ExteriorWall.setText("")
self.labelEnvironmental_ExteriorWall.setAlignment(QtCore.Qt.AlignCenter)
self.labelEnvironmental_ExteriorWall.setObjectName("labelEnvironmental_ExteriorWall")
self.layoutExteriorWall.addWidget(self.labelEnvironmental_ExteriorWall)
self.labelEconomic_ExteriorWall = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelEconomic_ExteriorWall.sizePolicy().hasHeightForWidth())
self.labelEconomic_ExteriorWall.setSizePolicy(sizePolicy)
self.labelEconomic_ExteriorWall.setMinimumSize(QtCore.QSize(150, 20))
self.labelEconomic_ExteriorWall.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelEconomic_ExteriorWall.setFont(font)
self.labelEconomic_ExteriorWall.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelEconomic_ExteriorWall.setAlignment(QtCore.Qt.AlignCenter)
self.labelEconomic_ExteriorWall.setObjectName("labelEconomic_ExteriorWall")
self.layoutExteriorWall.addWidget(self.labelEconomic_ExteriorWall)
self.labelOverall_ExteriorWall = QtGui.QLabel(self.mainapp)
sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.labelOverall_ExteriorWall.sizePolicy().hasHeightForWidth())
self.labelOverall_ExteriorWall.setSizePolicy(sizePolicy)
self.labelOverall_ExteriorWall.setMinimumSize(QtCore.QSize(150, 20))
self.labelOverall_ExteriorWall.setMaximumSize(QtCore.QSize(100, 20))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.labelOverall_ExteriorWall.setFont(font)
self.labelOverall_ExteriorWall.setStyleSheet("background-color: rgb(255, 255, 255);")
self.labelOverall_ExteriorWall.setText("")
self.labelOverall_ExteriorWall.setAlignment(QtCore.Qt.AlignCenter)
self.labelOverall_ExteriorWall.setObjectName("labelOverall_ExteriorWall")
self.layoutExteriorWall.addWidget(self.labelOverall_ExteriorWall)
self.layoutOptimalSolution.addLayout(self.layoutExteriorWall)
self.line_20 = QtGui.QFrame(self.mainapp)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(11)
self.line_20.setFont(font)
self.line_20.setFrameShadow(QtGui.QFrame.Plain)
self.line_20.setFrameShape(QtGui.QFrame.HLine)
self.line_20.setObjectName("line_20")
self.layoutOptimalSolution.addWidget(self.line_20)
self.layoutRightBottom.addLayout(self.layoutOptimalSolution)
spacerItem22 = QtGui.QSpacerItem(0, 0, QtGui.QSizePolicy.Expanding, QtGui.QSizePolicy.Minimum)
self.layoutRightBottom.addItem(spacerItem22)
self.layoutRightBar.addLayout(self.layoutRightBottom)
self.gridLayout.addLayout(self.layoutRightBar, 0, 1, 1, 1)
MainWindow.setCentralWidget(self.mainapp)
self.menubar = QtGui.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 1118, 25))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(12)
self.menubar.setFont(font)
self.menubar.setObjectName("menubar")
self.menuFile = QtGui.QMenu(self.menubar)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(12)
self.menuFile.setFont(font)
self.menuFile.setObjectName("menuFile")

```

```

MainWindow.setMenuBar(self.menubar)
self.statusbar = QtGui.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)
self.actionOpen = QtGui.QAction(MainWindow)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(12)
self.actionOpen.setFont(font)
self.actionOpen.setObjectName("actionOpen")
self.actionClose = QtGui.QAction(MainWindow)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(12)
self.actionClose.setFont(font)
self.actionClose.setObjectName("actionClose")
self.actionSave = QtGui.QAction(MainWindow)
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(12)
self.actionSave.setFont(font)
self.actionSave.setObjectName("actionSave")
self.menuFile.addAction(self.actionOpen)
self.menuFile.addAction(self.actionSave)
self.menuFile.addAction(self.actionClose)
self.menubar.addAction(self.menuFile.menuAction())

self.retranslateUi(MainWindow)
self.tabWidget.setCurrentIndex(0)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "LEICAS - Life-cycle Environmental Impact and Costing Assessment"))
    self.labelInput.setText(_translate("MainWindow", "Input"))
    self.labelBuildingType.setText(_translate("MainWindow", "Building type:"))
    self.buildingTypeChoice.setItemText(0, _translate("MainWindow", "Choose..."))
    self.buildingTypeChoice.setItemText(1, _translate("MainWindow", "Residential"))
    self.buildingTypeChoice.setItemText(2, _translate("MainWindow", "Utilitarian"))
    self.labelCalculationChoice.setText(_translate("MainWindow", "Calculation Choice"))
    self.calculationTypeChoice.setItemText(0, _translate("MainWindow", "Choose..."))
    self.calculationTypeChoice.setItemText(1, _translate("MainWindow", "Generic"))
    self.calculationTypeChoice.setItemText(2, _translate("MainWindow", "Specific"))
    self.labelbuildingSystem.setText(_translate("MainWindow", "Building System"))
    self.buildingSystem.setItemText(0, _translate("MainWindow", "All"))
    self.buildingSystem.setItemText(1, _translate("MainWindow", "Bricks and concrete"))
    self.buildingSystem.setItemText(2, _translate("MainWindow", "Wood"))
    self.labelIFCFile.setText(_translate("MainWindow", "IFC-file:"))
    self.btn_OpenIFC.setText(_translate("MainWindow", "..."))
    self.labelGFA.setText(_translate("MainWindow", "Gross floor area (GFA):"))
    self.GFAInput.setPlaceholderText(_translate("MainWindow", "Insert GFA"))
    self.labelGFA_unit.setText(_translate("MainWindow", "<html><head/><body><p align=\"right\">m<span style=\" vertical-align:super;\">2 </span></p></body></html>"))
    self.btn_Calculate.setText(_translate("MainWindow", "Calculate"))
    self.labelOverallResults.setText(_translate("MainWindow", "Overall Results"))
    self.labelEI_category.setText(_translate("MainWindow", "Category"))
    self.labelEI_impact.setText(_translate("MainWindow", "Impact"))
    self.labelEI_unit.setText(_translate("MainWindow", "Unit"))
    self.labelEI_SC.setText(_translate("MainWindow", "SC"))
    self.labelGWP_name.setToolTip(_translate("MainWindow", "Global Warming Potential"))
    self.labelGWP_name.setText(_translate("MainWindow", "GWP"))
    self.labelGWP_unit.setText(_translate("MainWindow", "<html><head/><body><p>kg CO<span style=\" vertical-align:sub;\">2</span> eq</p></body></html>"))
    self.labelODP_name.setToolTip(_translate("MainWindow", "Stratospheric ozone depletion potential"))
    self.labelODP_name.setText(_translate("MainWindow", "ODP"))
    self.labelODP_unit.setText(_translate("MainWindow", "<html><head/><body><p>kg CFK-11 eq</p></body></html>"))
    self.labelPOCP_name.setToolTip(_translate("MainWindow", "Photochemical Oxidation Potential"))
    self.labelPOCP_name.setText(_translate("MainWindow", "POCP"))
    self.labelPOCP_unit.setText(_translate("MainWindow", "<html><head/><body><p>kg C<span style=\" vertical-align:sub;\">2</span></p></body></html>"))
    self.labelAP_name.setToolTip(_translate("MainWindow", "Acidification Potential"))

```

```

self.labelAP_name.setText(_translate("MainWindow", " AP"))
self.labelAP_unit.setText(_translate("MainWindow", "<html><head></body><p>kg SO<span style=\" vertical-align:sub;\">2</span>eq</p></body></html>"))
self.labelEP_name.setTooltip(_translate("MainWindow", "Eutrophication Potential"))
self.labelEP_name.setText(_translate("MainWindow", " EP"))
self.labelEP_unit.setText(_translate("MainWindow", "<html><head></body><p>kg PO<span style=\" vertical-align:sub;\">4</span>eq</p></body></html>"))
self.labelHTP_name.setTooltip(_translate("MainWindow", "Human Toxicity Potential"))
self.labelHTP_name.setText(_translate("MainWindow", " HTP"))
self.labelHTP_unit.setText(_translate("MainWindow", "1.4-DCB eq"))
self.labelTETP_name.setTooltip(_translate("MainWindow", "Terrestrial Ecotoxicity Potential"))
self.labelTETP_name.setText(_translate("MainWindow", " TETP"))
self.labelTETP_unit.setText(_translate("MainWindow", "1.4-DCB eq"))
self.labelFAETP_name.setTooltip(_translate("MainWindow", "Freshwater Aquatic Ecotoxicity Potential"))
self.labelFAETP_name.setText(_translate("MainWindow", " FAETP"))
self.labelFAETP_unit.setText(_translate("MainWindow", "1.4-DCB eq"))
self.labelMAETP_name.setTooltip(_translate("MainWindow", "Marine Aquatic Ecotoxicity Potential"))
self.labelMAETP_name.setText(_translate("MainWindow", " MAETP"))
self.labelMAETP_unit.setText(_translate("MainWindow", "1.4-DCB eq"))
self.labelADPc_name.setTooltip(_translate("MainWindow", "Abiotic Depletion Potential of Composites and Minerals"))
self.labelADPc_name.setText(_translate("MainWindow", " ADPc"))
self.labelADPc_unit.setText(_translate("MainWindow", "kg Sb eq"))
self.labelADPf_name.setTooltip(_translate("MainWindow", "Abiotic Depletion Potential of Fossil Fuels"))
self.labelADPf_name.setText(_translate("MainWindow", " ADPf"))
self.labelADPf_unit.setText(_translate("MainWindow", "kg Sb eq"))
self.labelTotalEI_name.setText(_translate("MainWindow", " Total"))
self.labelBC2012_name.setTooltip(_translate("MainWindow", "Building Decree 2012"))
self.labelBC2012_name.setText(_translate("MainWindow", " Building Decree 2012"))
self.labelBC2012_ApprovedDeclined.setTooltip(_translate("MainWindow", "Article 5.9 of the Dutch Building Decree 2012"))
self.labelBC2012_SCFinal.setTooltip(_translate("MainWindow", "m2 GFA / year"))
self.labelClassified.setText(_translate("MainWindow", "Generic"))
self.labelSpecif.setText(_translate("MainWindow", "Specific"))
self.labelCost_category.setText(_translate("MainWindow", " Category"))
self.labelCost_impact.setText(_translate("MainWindow", "Cost"))
self.labelSurface_Classified.setText(_translate("MainWindow", "Surface"))
self.labelCost_impact_2.setText(_translate("MainWindow", "Cost"))
self.labelSurfaceSpecific.setText(_translate("MainWindow", "Surface"))
self.labelCostInteriorWall_name.setTooltip(_translate("MainWindow", "Wall costs"))
self.labelCostInteriorWall_name.setText(_translate("MainWindow", " Interior Wall"))
self.labelCostFloor_name.setTooltip(_translate("MainWindow", "Floor costs"))
self.labelCostFloor_name.setText(_translate("MainWindow", " Floor"))
self.labelCostExteriorWall_name.setTooltip(_translate("MainWindow", "Wall costs"))
self.labelCostExteriorWall_name.setText(_translate("MainWindow", " Exterior Wall"))
self.labelTotalCost_name.setText(_translate("MainWindow", " Total"))
self.tabWidget.setTabText(self.tabWidget.indexOf(self.overviewTab), _translate("MainWindow", "Overview"))
self.labelImportanceofImpact.setText(_translate("MainWindow", "Importance of impact"))
self.environmentalImpact.setText(_translate("MainWindow", "Environmental Impact"))
self.economicImpact.setText(_translate("MainWindow", "Economic Impact"))
self.btn_Generate.setText(_translate("MainWindow", "Generate"))
self.labelOptimalSolution_name.setText(_translate("MainWindow", "Best Choice Element Types"))
self.labelElements.setText(_translate("MainWindow", " Elements"))
self.labelEnvironmental.setText(_translate("MainWindow", "Environmental"))
self.labelEconomic.setText(_translate("MainWindow", "Economic"))
self.labelOverallSolution.setText(_translate("MainWindow", "Overall"))
self.labelName_InteriorWall.setTooltip(_translate("MainWindow", "Global Warming Potential"))
self.labelName_InteriorWall.setText(_translate("MainWindow", " Interior Wall"))
self.labelEconomic_InteriorWall.setText(_translate("MainWindow", "<html><head></body><p><br></p></body></html>"))
self.labelName_Floor.setTooltip(_translate("MainWindow", "Acidification Potential"))
self.labelName_Floor.setText(_translate("MainWindow", " Floor"))
self.labelEconomic_Floor.setText(_translate("MainWindow", "<html><head></body><p><br></p></body></html>"))
self.labelName_ExteriorWall.setTooltip(_translate("MainWindow", "Stratospheric ozone depletion potential"))
self.labelName_ExteriorWall.setText(_translate("MainWindow", " Exterior Wall"))
self.labelEconomic_ExteriorWall.setText(_translate("MainWindow", "<html><head></body><p><br></p></body></html>"))
self.menuFile.setTitle(_translate("MainWindow", "File"))
self.actionOpen.setText(_translate("MainWindow", "Open"))
self.actionOpen.setShortcut(_translate("MainWindow", "Ctrl+O"))
self.actionClose.setText(_translate("MainWindow", "Close"))
self.actionClose.setTooltip(_translate("MainWindow", "Close"))
self.actionClose.setShortcut(_translate("MainWindow", "Esc"))
self.actionSave.setText(_translate("MainWindow", "Save"))
self.actionSave.setShortcut(_translate("MainWindow", "Ctrl+S"))

```

```
if __name__ == "__main__":  
    import sys  
    app = QtGui.QApplication(sys.argv)  
    MainWindow = QtGui.QMainWindow()  
    ui = Ui_MainWindow()  
    ui.setupUi(MainWindow)  
    MainWindow.show()  
    sys.exit(app.exec_())
```

APPENDIX IV - ENVIRONMENTAL PRODUCT DECLARATIONS

The thought of calculating the embodied energy of a building is currently remarked as a driver for the determination of the total energy required during the life-cycle of a building (Curran, 2013). Additionally, the choice of materials is also being considered as being significant in the LCA of buildings (van Gemert, 2019). Previous researches identify the embodied impacts of building materials on three levels: (i) individual building materials (material level); (ii) collections of materials (product level); and (iii) the whole building (building level). Within these studies, a variety of functional units are used which are found to be based on whole buildings or an m² of floor area (Cabeza, Rincón, Vilariño, Pérez, & Castell, 2014). Other researches define that their functional unit is based upon the different building elements or the weight of materials (Anand & Amor, 2017). Contributors to the AEC-industry such as manufacturers, construction firms, and designers are demanding information that will enable them to make environmental decisions which address the environmental impacts of materials and products in buildings and other construction works. Therefore, an answer to this need is to create environmental product declarations (EPD). These are environmental declarations providing quantified environmental data for (pre)determined indicators using LCA. EPDs offer quantified environmental information on the life-cycle of products. In Europe, EPD standards are based upon the requirements of EN 15804, which provides fundamental rules for the product category of construction products. In 2012, the European Committee published the EN 15804, which is part of a suite of standards for the assessment of the sustainability of construction, which works at both product level as well as building level. This suite of standards includes:

- EN 15643-1, Sustainability of construction works – Sustainability assessment of buildings – Part 1: General framework;
- EN 15643-2, Sustainability of construction works – Assessment of buildings – Part 2: Framework for the assessment of environmental performance;
- EN 15978, Sustainability of construction works – Assessment of environmental performance of buildings – Calculation method;
- CEN/TR 15941, Sustainability of construction works – Environmental product declarations – Methodology for the selection and use of generic data;
- EN 15942, Sustainability of construction works – Environmental product declarations – Communication format: business to business.

Since 2013, the Dutch government has imposed an obligation to make an environmental performance calculation by request for a building application. However, this did not yet include a limit value as a minimum requirement. From January 2018 this limit value has been set at a maximum of 1, which means that building applications submitted after 1 January 2018 must be provided with an environmental performance calculation with a maximum result of 1. The score of the environmental performance calculation is based upon the guidelines of the '*Bepalingsmethode Milieuprestatie gebouwen en GWW-werken*', a framework for calculating environmental impacts of individual building products (EN 15978).

APPENDIX V - THE RESULTS OF THE DIFFERENT VARIATIONS OF THE CASE STUDY

GENERIC CALCULATION CHOICE - BUILDING SYSTEM ALL



Figure 27 - Snippet with results of the generic scenario based on building system "All"

SPECIFIC CALCULATION CHOICE - BUILDING SYSTEM ALL



Figure 28 - Snippet with results of the specific scenario based on building system "All"

GENERIC CALCULATION CHOICE - BUILDING SYSTEM BRICKS/CONCRETE

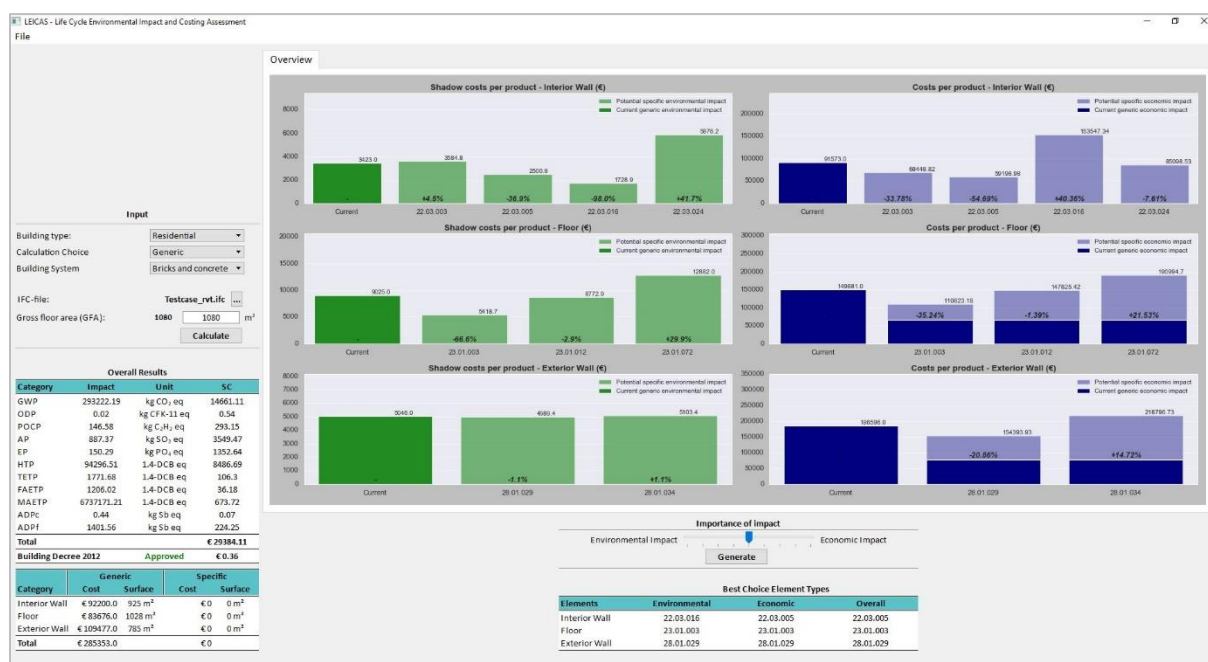


Figure 29 - Snippet with results of the generic scenario based on building system "Bricks & concrete"

SPECIFIC CALCULATION CHOICE - BUILDING SYSTEM BRICKS/CONCRETE

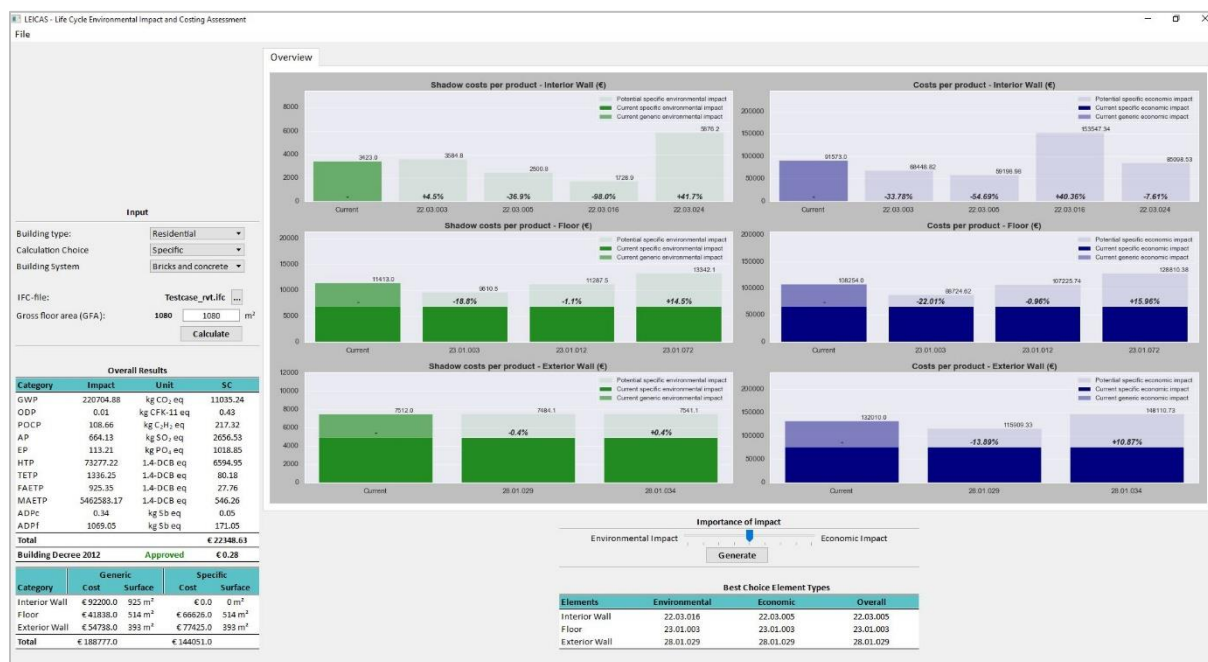


Figure 30 - Snippet with results of the specific scenario based on building system "Bricks & concrete"

GENERIC CALCULATION CHOICE - BUILDING SYSTEM WOOD

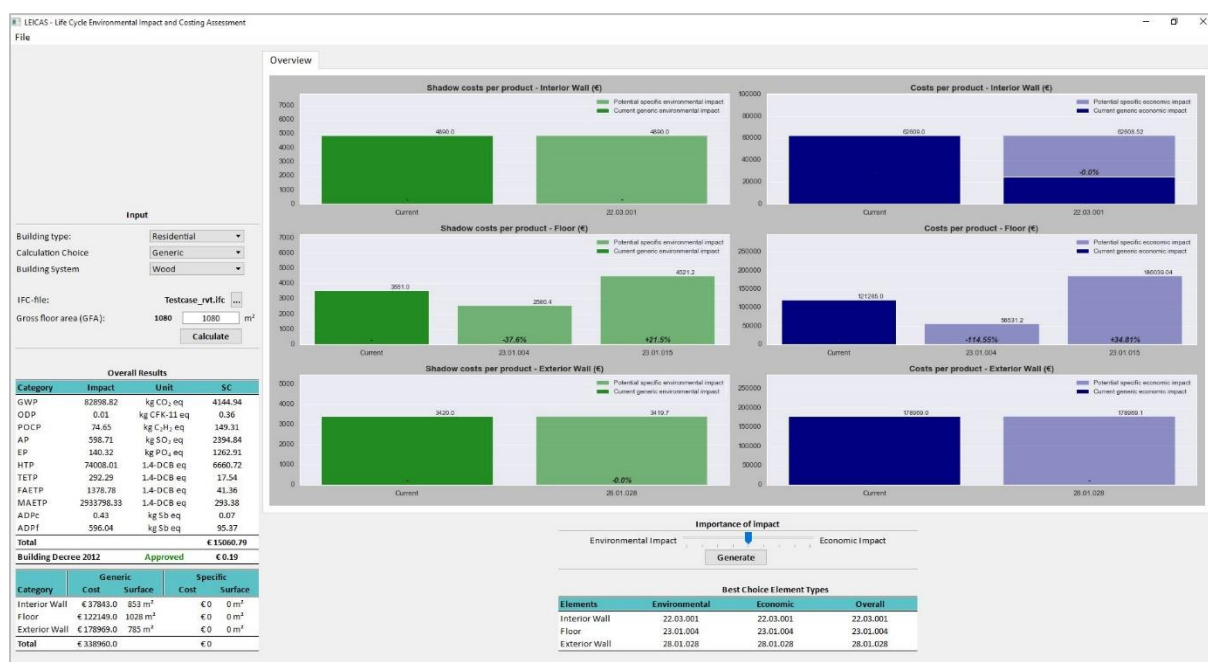


Figure 31 - Snippet with results of the generic scenario based on building system “Wood”

SPECIFIC CALCULATION CHOICE - BUILDING SYSTEM WOOD

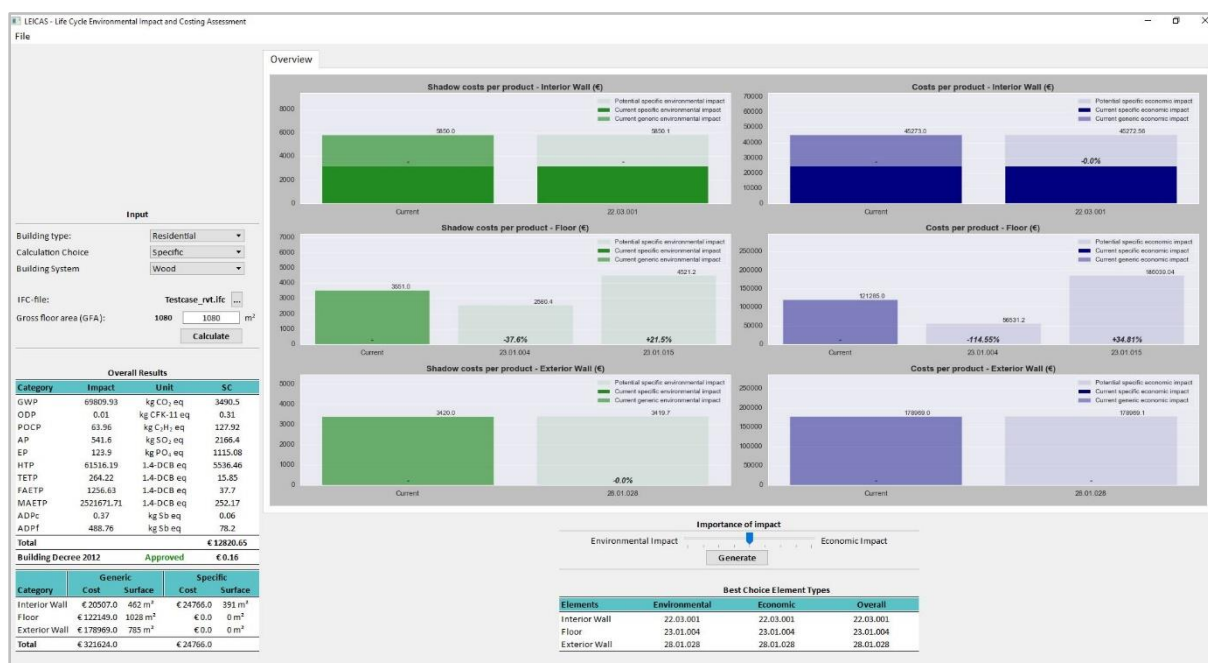


Figure 32 - Snippet with results of the specific scenario based on building system “Wood”