BIM-BASED INVESTIGATION OF TOTAL ENERGY CONSUMPTION IN DELIVERING BUILDINGS AS A PRODUCT

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by RAJA SHAHMIR NIZAM SHAIKH

FEBRUARY

2019

Table of Contents

ABSTRACT

Considerable efforts have been made to reduce buildings' operational energy use over the last decades, but little attention has been paid to reduce the material, transportation and construction energy. Focusing only on the operation phase forgoes the opportunity to reduce other building-related energy consumption, and even if the environmental impacts arising from construction and transportation are small as compared to the operation phases, its cumulative impact at the national level is of concern. Assessment of the energy associated with the material production, transportation and construction of buildings provides an opportunity for reducing the use of energy and improving sustainability. Building Information Modeling (BIM) provides a platform to incorporate sustainability information in the design of buildings. However, interoperability of BIM with Life Cycle Assessment (LCA) tools needs further investigation.

Previous research in this area has either partially employed BIM; data was exported from the main BIM authoring tool and then auxiliary tools were utilized to evaluate the model, thereby causing a disconnect between the model and analysis resulting in non-interoperable systems, or has ignored the importance of retaining the LCA results within the BIM environment. To address this issue, this study presents a framework to estimate the embodied energy content within the BIM environment. The implementation of this framework is illustrated by the development of a tool for estimation of material embodied energy, transportation energy and construction energy. Simulation models are created, which can be used as templates for energy optimization during transportation and construction.

By analyzing different resource combination scenarios, lower energy consumption can be achieved.

The presence of embodied energy results gives way to the potential of sensitivity analysis to optimize the embodied energy content. Therefore, there is a need for a formal and standard definition of embodied energy content in BIM. Such inclusion is necessary to perform related model sharing through main stream BIM data exchange protocols such as Industry Foundation Classes (IFC). Therefore, this study also proposes an extension of the IFC model to incorporate embodied energy information. Matching the aspects of cost associated with the building elements and processes, a new resource of energy is proposed. Consequently, the abstraction of this energy resource is further embedded by establishing the relationship with other model entities. A prototype tool has been implemented on a case project to establish the workability of the framework.

ACKNOWLEDGEMENT

In the name of Allah, the most merciful and the most benevolent!

I would like to pay my greatest appreciation to my supervisor Dr. Cheng Zhang, for her utmost kindness, support, encourangement and advice. She always kept a balanced and empathetic approach towards the circumstances, which enabled me to work in a calm and productive environment, for which I remain grateful to her.

I would like to thank my mother, Mrs. Zuhra Nizam (may her soul rest in peace) for her love, care and tolerance. She passed away during the third year of my PhD. I have never met a person better than her.

I would also like to thank my father, Prof. Nizamuddin Shaikh. It is only due to him that I pursued PhD. I have never met a person more knowledgable than him. Despite his old age and weakening body, his encouragements are always at par.

Special thanks are due to my wife (Engr. Maria), her love, patience and support keep me going in the rainy days. Again, I feel extremely lucky to have a life partner who completely understands the concerns of a researcher and handle the circumstances carefully.

Appreciation is due to Dushu Lake Higher Education Town for granting me a much needed scholarship. Finally, I would like to thank Xian Jiaotong Liverpool University and all the staff of Department of Civil Engineering for letting me work in their space.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Background

Energy and its related problems are a worldwide concern. According to the International Energy Agency (IEA), the total primary energy supply (TPES) reached a historical high record of 13,113 million tons of coal equivalent (Mtce) in 2011, releasing 31,342 million tons of energy related CO_2 emissions, which is regarded as the major human cause of global warming [1]. China, as a major contributor, accounted for approximately one-fifth of the global TPES and one-fourth of CO_2 emissions. The Intergovernmental Panel on Climate Change (IPCC) reported that the building sector was responsible for 40% of global energy consumption and approximately 25% of global CO_2 emissions [2]. The construction industry is a highly active industry in both developed and developing countries; therefore, the social, economic and environmental indicators of sustainable development are drawing more attention. Especially in China, the rapid development of the economy warranted extensive construction work.

Speaking optimistically, however, buildings may also provide an opportunity to reduce this burden by implementing strategies to reduce energy consumption. Therefore, aiming for most sustainably constructed building by using limited additional resources has become a prime goal for most construction projects. Energy being an important part of the sustainability drive, usually takes a lead role in determining the achievement of the sustainability goal. Buildings are associated with two types of energy consumption, embodied energy and operational energy. Embodied energy encompasses all the energy that is consumed before the operation phase of a building i.e. energy consumed during material manufacture, transportation and construction [3,4]. Considerable efforts have been made to reduce buildings' operational energy use over the last decades, but little attention has been paid to reducing the construction energy use, i.e. the energy consumed during the off-site production and transportation of building materials and components, the erection of the building on-site, and any refurbishment activities that may be required. Focusing only on the usage phase forgoes the opportunity to reduce other building-related emissions, and even if the environmental impacts from construction are small compared to other phases, these impacts can be large when looked at on a national level.

Construction projects are becoming more complex and difficult to manage and as technology develops, more construction professionals are familiarizing themselves with Building Information Modelling (BIM). The BIM technique is relatively new and has been developed extensively in recent years. BIM is described as a data modelling technique which integrates the information of different domains for a given engineering project. BIM is mainly used in the design stage, construction stage, and late operation and management stages. It can considerably improve efficiency and reduce risk throughout the entire constructional engineering process. There are a number of views in industry and academia as to what constitutes BIM. An abundance of definitions related to BIM has emerged with terms including object-oriented modelling, project modelling, virtual design and construction, virtual prototyping, integrated project databases and the more recent term BIM [5]. In Architecture, Engineering and Construction (AEC) industry, design and construction activity involves numerous organizations working together and depending on each other to provide tailored solutions for owners. BIM has emerged into the mainstream bringing a different process of collaboration and a new way of working transforming current AEC industry structures and practices, with the aim of improving efficiency & environmental objectives [6].

BIM is currently the most common denomination for a new way of platforming for data integration, which enables a lot of analysis to be done more efficiently, one good example is a study conducted by Abanda et.al [7], where data from different sources such as energy consumption during material procurement was estimated using BIM. The underlying data structure of BIM models is based on Industry Foundation Classes (IFC) data model. IFC is a set of definitions describing the consistent data representation of building components [8]. IFC describes an EXPRESS based entity-relationship model consisting of hundreds of entities ordered into an object-based heritage hierarchy.

To manage the data of energy consumption and evaluate the alternatives needs an intelligent repository that can hold this data and conduct meaningful extraction of data as well as perform simulations. BIM has the ability to perform these critical tasks with some support of external database systems. BIM models usually contain the geometry of the building as well as the associated properties of the building elements. However, it does not contain information such as embodied content or construction energy of the building. Nevertheless, BIM authoring tools are flexible enough to hold additional information and perform calculations on the combination of additional and built-in information [9].

1.2 Problem Statement

The performance of construction projects is recently being tied with sustainability along with time and cost. Numerous sophisticated tools are available to control the traditional factors of cost and time. However, sustainability has not yet been properly quantified, and a lot of research is being done to quantify the various aspects of sustainability. Embodied energy content is one such parameter that can indicate the sustainable quotient of the building. However, the estimation of embodied energy and its application towards the evaluation of sustainable alternatives face several challenges. The first challenge is the absence of a strategy to estimate embodied energy at the design stage incorporating foreseeable transportation and construction impact. This absence provides a hinderance in consideration of sustainability as a criterion for selection of project alternatives. The second challenge is the lack of a convenient tool to comprehensively input all relevant data and link the construction process with the embodied energy estimation which includes establishing a relationship with external data to obtain meaningful, quick and comparable solutions for embodied energy. The third challenge is the integration of embodied energy in the underlying data structure of the BIM model. Since cost had been a critical factor for assessing project performance at the time when the IFC schema was developed, therefore, cost was adequately mapped within the IFC schema. Albeit sustainability parameters are also mapped in the IFC schema, but their effectiveness is not adequate, because unlike cost, the intention was just to set the physical properties of materials rather than input for decision-making scenarios.

BIM provides a platform to integrate all such information in a single environment and enable further development to analyze the building data for various applications. Previous efforts in this research domain were not able to realize the full potential of BIM in relation to sustainability assessment as most studies exported the model from the main BIM authoring tool and do not consider bringing back the results to the BIM model thereby making the effort useless for incorporating changes in the model. Furthermore, there has been limited consideration regarding the embedment of embodied energy data in the IFC model.

In the light of the above discussion, this study provides a path in an effort to overcome the above-mentioned challenges in several ways. Primarily, this study intends to enhance the potential of BIM by keeping the BIM authorized tool at the center of its methodology so that the results of the analysis are available in the BIM environment for further refinement. Secondly, attribution of LCA data is automatic and does not require any further manual acknowledgement and the construction phase is fully integrated into the proposed approach by realizing energy consumption of construction equipment as well as making the energy data available for simulation-based optimization. Thirdly, the user interface is kept simple, as well as engaging, and minimal effort is required to obtain meaningful results in the native environment. Finally, an extension to the IFC is proposed to embed the embodied energy information. The capability of the tool is made consistent with the extended IFC to create IFC files with embodied energy information which enhances the interoperability of energy data.

1.3 Aim and Objectives

The aim of the study is to propose a practical and workable solution for considering sustainability as a critical factor for the evaluation of building as a product before delivering to the owner. Embodied energy content is considered as the sustainability quotient of the building. Following objectives are set to achieve the aim:

1. Investigate the total energy consumption in delivering building as a product.

2. Propose a method to automatically calculate the energy consumption including embodied energy in material, energy consumption in transportation and energy consumption in construction.
3. Provide possible suggestions for energy saving based on construction process analysis based on simulation.
4. Enhance the interoperability of sustainability data by extending the current IFC model to incorporate embodied energy information.

1.4 Organization of Thesis

The rest of this thesis is organized as follows: Chapter 2 contains a literature review around the estimation of embodied energy through BIM and embedding embodied energy information in IFC. Chapter 3 discusses the methodology of this study comprising of data collection, data mapping, data analysis, simulation models and IFC embedment. Chapter 4 elaborates the system prototype development including the details of the application programming interface and database structures. Chapter 5 illustrates the implementation of the prototype in terms of the workflow of the prototype and data requirements. Chapter 6 demonstrates a case study employing the developed prototype for estimation and optimization of embodied energy content of a building as well as embedding the results in an IFC file. Finally, Chapter 7 presents a conclusion for this project by shedding light on the outcomes of the project and how the objectives are met to achieve the aim.

CHAPTER 2: LITERATURE REVIEW

This study tends to automate the estimation and optimization of embodied energy through BIM and thus demands a review around six domains and their interrelation. The first domain is the embodied energy with respect to the material intensity, a review of the studies which employed various material energy databases in order to estimate a building's material energy component is presented in section 2.1. The second domain comprises of the embodied energy with respect to transportation, a review of the studies which considered the estimation of material transportation is presented in the section 2.2. The third domain targets the literature around construction energy, mainly studies focused on the energy consumption by construction machinery and equipment. And the fourth domain discusses the evolution of BIM emphasizing on the sustainability aspect of BIM in general and estimation of environmental impact through BIM in particular. The fifth domain is focused on the optimization of energy consumption through construction and transportation simulation, section 2.5 presents a review of the different construction and transportation process models adopted in various studies. The sixth domain investigates the IFC data structure and comprises of studies involving the extension of IFC model. Finally, a summary section is planted to explain the interrelation of these six domains and thus, establish a background for the methodology section.

2.1 Embodied Energy in Material

Building energy is divided into two broad categories: embodied energy and operational energy [3]. Many studies have been conducted to optimize operational energy, but quantification and optimization of embodied energy has not been extensively considered [10]. Embodied energy entails energy consumption excluding the operation phase i.e., energy consumed during material extraction, transportation and installation on site [11]. Initial and recurring embodied energy are the two major components of embodied energy. Initial embodied energy is the sum of the energy required for extraction and manufacture of a material together with the energy required for transportation of a material used for the initial building construction. The recurring embodied energy in buildings represents the sum of the energy embodied in the material use due to maintenance, repair, restoration, refurbishment or replacement during the service life of the building [12]. However, the recurring embodied energy is a small percentage of the total embodied energy and therefore is neglected. So, this study refers to the initial embodied energy as the embodied energy throughout this report.

Simplistically, the energy required in obtaining the materials from their rawest form, transportation of the material to the site and finally its assembly in the form of a structure is referred to as embodied energy. To quantify this energy is almost a challenge in itself, but a bigger challenge is to generate realistic alternatives based on embodied energy and simulate these alternatives to find the alternative with optimum embodied energy.

The methodology of Life Cycle Assessment (LCA) is adopted to obtain the embodied energy. Process Analysis, input-output(I/O) and hybrid analysis are the three approaches used in LCA. Although process analysis is the widely used approach, the complex relationship between the goods and services causes a substantial disadvantage in using this approach. On the other hand, economy based national average data is used by I/O analysis which is regarded as a more comprehensive approach by some researchers. Han et al. [4] presented a detailed embodied energy consumption evaluation framework for building construction engineering. They presented that the embodied energy consumption of construction engineering industry is quantified as 7.15E+14 J, and the sum of the embodied energy by steel, cement, lime and metal products is more than 3/4 of the total embodied energy consumption. The database used in their research is the Chinese economy 2007, which is built up based on an input-output modeling, and includes six sources for direct external energy inputs which are divided into two groups as fossil sources (coal, crude oil, and natural gas) and non-fossil sources (hydropower, nuclear power and firewood). The third approach which is a hybrid of the above mentioned two approaches was adopted by Inventory of Carbon and Energy (ICE) database [13]. This database lists different coefficients to represent the impact a certain material brings in terms of embodied energy and carbon emissions. Chau et al. [14] noted that for virgin and recyclable building materials manufactured in China, there is an absence of a comprehensive database of embodied energy coefficients. However, for the places which lack such databases, ICE is the most favorable choice [15]. This is because when ICE values were correlated with those calculated by using four different approaches (Conventional Input-Output based, Basic Input Output Hybrid model with actual energy use integration, Input Output Hybrid model with human and capital energy and Input Output Hybrid model with sectoral disaggregation) in the country where embodied energy data was present, a very strong positive correlation ($r^2 = 0.9$) was found.

Especially in the context of China, many studies opted ICE as the embodied energy and carbon emission coefficients data source. Such a study [16], used ICE to evaluate the sustainability aspect of a residential structure in the Chinese context. Similarly [14], calculated the embodied energy for a high-rise office building in

Hong Kong using ICE, where most of the building materials were imported from China. Therefore, the Inventory of Carbon & Energy database is used to obtain the embodied energy coefficients in this study. However, the information provided by ICE is usually limited to extraction or production of the material, transportation and construction aspects are missing [17].

2.2 Energy Consumption in Transportation

Transportation has a considerable impact on the environmental and air quality on construction activities. The case study done by Monahan and Powell [18] has showed that transportation from factories to the site resulting in 2% of the total embodied carbon. In another study conducted by Moncaster and Symons [19], they have shown that the stage of transportation contributes 9% and 10% of total embodied carbon and total energy respectively for a typical masonry residential unit. For a project specific evaluation, transportation impacts can be calculated by incorporating hauling distances, vehicle efficiencies and the heating value of fuel. Artenian et al. [20] reported that the transportation sector is one of the largest contributors to global Green House Gas (GHG) emission, where approximately 73% of these emissions are generated by road transportation and freight related traffic due to construction.

By observing case studies from different geographic locations but same I-O model, the transportation energy contribution remains 3 to 4%. A case study conducted by Chang et. al [21] for high-rise buildings in China concluded that the contribution of transportation energy is around 4%. Similarly, Nässén et al. [22] studied 18 buildings in Sweden and Denmark using the I-O model and estimated the contribution of 3% on average. Treloar [23] laid outa methodology for estimating transportation energy using the hybrid I-O model which was later adopted by many studies such as Stephan and Stephan [24] to calculate the transportation energy component for the Lebanese residential buildings. Although the transportation phase was found to have less impact on the total embodied energy, however, for the sake of completeness transportation phase is integrated in the research product.

2.3 Energy Consumption in Construction

Guggemos and Horvath [25] have proposed an augmented process-based hybrid LCA model – the Construction Environmental Decision Support Tool (CEDST) - to analyze the environmental effects from the construction phase of commercial buildings as applied to a California building. The authors argued that significant larger use-phase effects often overshadow the construction-phase in building LCA. However, such construction-phase effects, when aggregated at the national level, may prove to be significant. CEDST evaluates environmental effects from the manufacture of temporary materials used in the construction process (e.g., form-work), transportation of materials and equipment, equipment use, and waste generation during construction. In the case study, equipment use accounted for about 50% of environmental effects and the major contributors are concrete mixer trucks, concrete pumps, cranes, and air compressors, while temporary construction materials had the second largest impact on the environment.

The findings of embodied energy consumption in construction by Moncaster and Symons [19] also concluded that the residential unit contributed 3% and 5% of the total embodied carbon and energy impacts, respectively. Sharrard et al. [26] have focused on the environmental and energy implication of the construction process, specifically on-site energy consumption by creating a broader boundary for considering the energy use and environmental impacts of engines and vehicles used for construction activities. On-site diesel fuel is used to power bulldozers, excavators, cranes, generators, and other types of equipment. They have quantified how significant construction is in terms of energy usage, which used 2.5–2.9 PJ of energy annually, and accounted for 2.6–3% of the entire energy consumption of the US.

Crawford and Trealor [27] laid out a methodology to estimate the construction energy by modifying the method provided by Stephan and Crawford [28]. This method was completely based on the hybrid energy coefficients which in turn required the economic transaction information with respect to the output sector. However, Zhao and Li [29] outlined a process-based methodology for quantitatively assessing the environmental impact of construction activities in terms of energy and carbon emission by using different data sources. In this study, a similar method is adopted to assess the construction energy.

2.4 Building Information Modelling

BIM is one of the most promising developments in the Architecture, Engineering and Construction (AEC) industries. According to National BIM standards, “BIM is a digital representation of physical and functional characteristics of a facility and a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition” [30].

Many researchers have evaluated the effectiveness of BIM applications within different educational or industrial settings, BIM is accepted as a process and corresponding technology to improve the efficiency and effectiveness of delivering a project from inception to operation/maintenance. A number of case studies have been published that show useful BIM implementations on actual construction projects including the use of proposed use of 3D/4D models for design review from the perspective of constructability, design of a BIM-based game for fire evacuation simulations, 4D visualization technology for safety [31].

The key concept of BIM is a complete object-oriented digital model of the building that can be reused by different stakeholders at different stages of the building life cycle. Besides the geometric information, other building information that is not available in traditional computer-aided design (CAD) solutions is also stored in the model, including building materials and costs, project specification, and contract information. The stored information can be easily retrieved and reused once inputted into the model, thus eliminating the need to re-input or even recollect information that is valuable in later stages of the building's life cycle [32].

Many studies have been conducted for the BIM-LCA integration in general and for the estimation of embodied energy in particular. These studies can be categorized into four major types (Type I to Type IV) to understand their utility better and to realize their limitations. These types are not mutually exclusive i.e. a certain study may have the characteristics of more than one type. A description of each type along with its representative cases are illustrated in the following discussion. A comprehensive list of BIM-LCA tools is provided in the Table 2-1. The first type (Type I) includes studies which focus on project-based outcomes but lack the general framework that could be extended to other projects, such a study was conducted by Shrivastava & Chini [33], which assessed the initial embodied energy of a building through BIM. This study criticizes the export of BIM quantity take-offs to other platforms for sustainability analysis and proposes the extension of material properties to incorporate sustainability data within the BIM environment.

The manual input of material sustainability data into the material properties database makes this method impractical as it would increase modelling time and is prone to errors. Another study [34] analysed the embodied energy and CO2 emission of mud-brick and cement block houses. Each building element was treated separately, and manual calculations were performed to estimate the embodied energy. Then Revit was used to validate the results. However, a general framework for assessing embodied energy content of different building projects is missing in both the studies.

The second type (Type II) includes studies which treat BIM as just a quantity take-off tool and export the quantities to other software. Shadram et al. [35] proposed an integrated BIM-based framework for minimizing embodied energy during building design. Despite providing a comprehensive framework for estimation of embodied energy, this method lacks interoperability as it loses focus from BIM and uses Power Pivot as the main data integration platform. Likewise, another effort to conduct LCA through BIM is done by Jrade & Abdullah [36]. This method proposes to export the BIM model as an IFC file to a LCA tool [37].

The third type (Type III) includes studies which ignore the construction phase for the estimation of embodied energy. In other words, the boundary conditions of these studies are limited to the cradle to site. In their study Abanda et al. [38] thoroughly integrated BIM with UK new rules of measurement. This study was extended by Abanda et al. [7] to incorporate LCA data. However, this integration ignored the impact of the construction phase on the embodied energy. In addition, a famous commercial tool "Tally" [39] lies within this category. As shown in Figure 2-1 below, Tally takes lump sum input for construction energy and ignores the analysis of construction processes or equipment for conducting LCA. Tally also ignores the level of detail present in the model and requires attribution of LCA data for every material assignment irrespective of the information already detailed in the model.

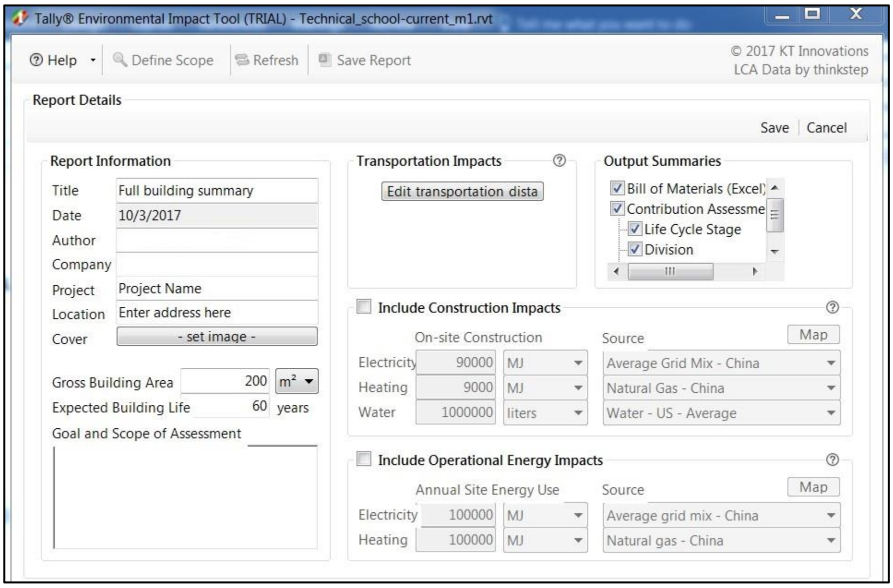


Figure 2-1: Tally user interface [39]

The fourth type (IV) of previous studies either provide inadequate detail of the process or provide complex and rigorous methods that may not seem practical enough to be adopted. Furthermore, their utility may be limited to any one aspect of the BIM-LCA integration process. There are several studies included in this category. Jrade & Jalaei [40] intended to incorporate sustainability performance within the BIM environment. The framework required to establish a database of sustainable materials with a unique keynote for every material. This keynote must be represented in every Revit material family according to the use (say building element type). The method seems impractical as it needs manual entry and matching of keynotes for numerous potential materials.

Another effort to perform LCA through BIM was done by Lee et. al. [41], which was focused on generating a green template for Korean construction industry. The framework needs to first create a Revit family (*.rfa) file through a particular family writer tool for every building element and then use that file for modelling in the BIM authoring tool (Revit). This not only increases the required skill level of the modeler but also make the built-in Revit family library unusable.

Table 2-1: List of BIM-LCA tools

Year Published	**Reference**	**Tools Used**	**Input/ Output**	**Limitations**	**Category**
2015	Ajayi et. al. [42]	Revit, Green Building Studio (GBS), ATHENA Impact Estimator, MS Excel	Input: BIM Model, Manual Entry in LCA tool. Output: Global Warming Potential (GWP), Human Health Impact (HHI)	-Relying on industry averages of the LCA tool for transportation and construction impact -The results are no longer available in the BIM environment	Type II
2014	Abanda et. al. [34]	Revit, MS Excel, ICE Database	Input: BIM Model, Excel templates for calculations, Manual linking of quantities with relevant coefficients. Output: Embodied energy and Embodied carbon emissions	-Transportation and construction impact was ignored - The results no longer available in the BIM environment -Methodology tailored for validation of a specific project	Type III Type II Type I
2016	Shadram et. al. [35]	Revit, Power Pivot, FME,	Input: BIM Model, EPD for building materials, Location	-Construction impact was ignored.	Type III Type IV Type II

		Google Maps API	of suppliers and construction site, Recipes for different building elements. Output: Embodied energy for Cradle to gate (A1 to A3) and gate to site (A4)	-Need to make a recipe for each kind of building element. -Need to convert EPDs manually from Pdfs and load in database by using a specific format. -The results are no longer available in the BIM environment	
2012	Jrade & Abdullah [36]	Revit, IFC Analyzer, MS Excel, Athena Impact Estimator,	Input: BIM Model, Manual Entry of Tags for Excel Manipulation. Output: Full range of LCA parameters available through Athena Impact Estimator	-Assumed industry average for all the impacts. -Work with very simple models. -Manual Export and Import from Revit, IFC File Analyzer, Excel and Athena Impact Estimator. -Results are no longer available in the BIM environment.	Type II Type IV
2017	Abanda et. al. [7]	Revit, Navisworks, Excel, Revit API	Input: BIM Model, Excel Templates, ICE Database, Navisworks manipulations. Output: Embodied energy and carbon (A1 to A3)	-Transportation and Construction impact is ignored. -Navisworks manipulation is required for each project but it is not integrated in the automated framework.	Type III Type II Type I

2012	Shrivastava & Chini[33]	Revit	Input: BIM Model, Manual entry of embodied energy values from Canadian Architect website. Output: Embodied energy (A1 to A3)	-Transportation and Construction impact is ignored. -Manual entry of embodied energy values in the additional shared parameter of building material family without linking back to any other Revit entity. -The framework limited to a single project, require repeated effort to perform the analysis on different machines	Type III Type IV Type I
2013	Jrade & Jalaei [40]	Revit, Athena Impact Estimator, Excel	Input: BIM Model, Separate Database for Sustainable building materials, Manual Entry for keynotes for each material family according to use in a particular building element. Output: Full range of LCA parameters available through Athena Impact Estimator, LEED Performance	-Relying on industry averages for Transportation and Construction impacts -Ample amount of preliminary work is required to make a database for sustainable materials and link with the keynotes entered in the BIM software. - Results are no longer available in the BIM environment.	Type IV Type II

2015	Lee et. al. [41]	Revit	Input: BIM model, Database for Korean Life-Cycle Inventory of materials, Revit Family Libraries to be used for modelling Output: Environmental impact of building materials in terms of GWP, AP, EP, ODP, POCP, ADP. (A1 to A3)	-Transportation and Construction impact is ignored or relied on industry averages without considering construction equipment and method used. -Workable for a specific region (Korea). - Revit family libraries consistent with the template are to be established before modelling.	Type III Type IV
2015	Peng [43]	Revit, Ecotect and Excel	Input: BIM Model, Emission Coefficients from previous studies, National grid averages for emission calculations Ouput: CO2 emissions for the whole lifecycle	-No method for estimation of construction impact is provided (Lumpsum values for construction energy are adopted from previous studies). -Relied on Industry standards for transportation impact. - Project based estimation with no general framework different projects - Results no longer available in the BIM environment.	Type III Type I Type II

2015	Shin & Cho [44]	ArchiCAD, Excel	Input: BIM Model, Emission Coefficients for Material Manufacturing, Construction and Transportation. Output: Equivalent Carbon Emissions for the whole lifecycle	-BIM software was only used to calculate the quantities. -No general framework for different projects -Ample amount of manual input - Results are no longer available in the BIM environment.	Type II Type I Type IV
	Tally[39]	Revit	Input: BIM Model Output: Complete LCA with comprehensive results	-Commercial Software; no further customized development on top of the tool is possible -Ignores construction detail and asks for one lumpsum value. - Results are no longer available in the BIM environment (final report presented in pdf or excel format and results inaccessible inside Revit)	Type III Type II

In summation, the previous studies conducted mainly lack in four broad areas: Focused on project-based outcomes (unfit for generalization), treat BIM just as a quantity take off tool, ignore the construction phase for estimation of embodied energy or propose complex and rigorous methods that seem impractical for routine projects.

2.5 Construction Simulation

The operational behavior of construction systems can be understood by representing it in the form of construction simulation models. These models are used to optimize various project goals such as minimize costs or project duration and help to improve overall construction project management [45]. Many researchers have adopted different approaches to model construction systems such as different quantitative models were proposed by Shena et al. [46] and Sihabuddin and Ariaratnam [47] to investigate the environmental performance such as emissions. However, these modeling approaches lack the ability to capture the uncertainty around various parameters such as productivity of labor or equipment and weather conditions. Apart from that, these models seem to treat different construction processes independently, ignoring the dependencies and time dependent evolution of different parameters such as changes in queues and waiting time during the project.

As suggested by Gonzáleza and Echavegurenb [48], an alternative approach is to use discrete event simulation (DES) which significantly overcomes the limitations of quantitative modelling approach. The primary advantage of DES models is that it represents the construction process with the event based approach which allows time based evolution of the system. It helps the DES models to cater complex systems with high uncertainty and remain fruitful for dynamic decision rules as well as relationships between different entities and resources [49]. The application of DES modeling enables to perform synchronized analysis of any construction process involving resources, energy, residuals and/or emissions and supports to obtain precise and characteristic models.

Another significance of the DES approach is that the construction managers can play with different scenarios to find optimal solutions for their project goals such low emissions or cost [50]. DES has been widely used during the recent decades to simulate the construction processes such as excavations, loading, hauling and dumping by a number of researchers [51,52]. However, its application in measuring the environmental performance of construction projects has been limited to only a few recent studies [53,54].

Being an interdisciplinary technique, numerous platform support DES, but the first DES based construction simulation tool "CYCLONE" was introduced by Halpin [55]. Many other simulation tools were later developed to model various construction operations, such as STROBOSCOPE [56], ETZStrobe [57], Simphony [58], and so on. One platform that has significantly contributed in DES is Simphony, packaged as Simphony.NET, is a visual discrete-event simulation package specialized for construction systems and developed at the Hole School of Construction Engineering and Management, University of Alberta, Canada [59]. Apart from being a free for education purposes software, it is being continually evolved and it has versions as recent as August, 2017.

2.6 Industry Foundation Classes (IFC)

BIM solutions create and operate on digital databases for collaboration, manage change throughout those databases so that a change to any part of the database is coordinated in all other parts, and capture and preserve information for reuse by additional industry-specific applications. Data integration of the Construction is the basic of BIM, although BIM technology solves how to build information models, a unified standard is also needed to connect the work of information exchange and sharing etc. Therefore, developing general data models or interoperability standards is become to one of the important tasks to realize information integration [60]. As a comprehensive international standard for BIM interoperability, IFC is a set of definitions describing the consistent data representation of building components [8].

Various systems can implement converters between the IFC schema and their native data models to export and import IFC instances to exchange information with each other. This approach has contributed to addressing the problem of data exchange and integration. In addition to the shared IFC data model itself however, the interoperability also depends on two other factors: a) The quality of implementations of IFC export-import converters; b) The quality of the instance modeling of buildings. The respective work is carried out by application implementers and domain end-users alike [32].

In 1994 Autodesk constituted an industry group to guide the company on the expansion of a set of C++ classes that could support integrated application development. Initially named the Industry Alliance for Interoperability and altered its title in 1997 to the International Alliance for Interoperability. The new Alliance was re-formed as a no profit industry managed organization, with the objective of distributing the IFC as a neutral AEC product model answering to the AEC building lifecycle. An additional name alteration happened in 2005, and the IFC specification is currently managed by buildingSMART [61]. The first IFC model (IFC 1.0) was published in January 1997 and an addendum to the latest release

(IFC4 Add1) is published in July 2015. IFC describes multiple file formats that may be used, associating various encodings of the similar underlying data [62]:

1. IFC-SPF is a text format defined by ISO 10303-21 ("STEP-File"), where each line typically consists of a single object record, and having file extension ".ifc". This is the most widely used IFC format, having the advantage of compact size yet readable text.

2. IFC-XML is an XML format defined by ISO 10303-28 ("STEP-XML"), having file extension ".ifcXML". This format is suitable for interoperability with XML tools and exchanging partial building models. Due to the large size of typical building models, this format is less common in practice.

3. IFC-ZIP is a ZIP compressed format consisting of an embedded IFC-SPF file and having file extension ".ifcZIP".

IFC has four layers, IFC 4 consists of the (i) Resource Layer, (ii) Core Layer with Kernel & Extensions, (iii) Interoperability Layer and (iv) Domain Layer. A detail of the contents is provided in the Table 2-2 below which is adapted from BuildingSmart Chapters [63]. Building geometry and material property information can be exported to the standard format such as the IFC compliant STEP (Standard for Exchange of Product Model Data) physical data file (ISO 10303-21) from a BIM authoring tool by using IFC data model. STEP physical file has a header section and a data section; the header section contains information about the company, name, authorizing person, time and date when the export was done, IFC version used, the application that exported the file etc. The data section contains all occurrences for the entities of the IFC specification. These occurrences have a distinctive (within the scope of a file) STEP ID, the entity type name and a list of explicit attributes [64].

IFC describes an EXPRESS based entity-relationship model consisting of hundreds of entities ordered into an object-based heritage hierarchy. Examples of entities comprise of basic constructs such as IfcCartesianPoint, geometry such as IfcExtrudedAreaSolid and building elements such as IfcWall. The main division of IFC entities is into rooted and non-rooted entities. This division is based on the concept of identity (having a Globally Unique Identifier (GUID). Rooted entities have identity along with attributes for name, description, and revision control when the non-rooted only exist if referenced from a rooted instance directly or indirectly. Three abstract concepts further subdivide the IfcRoot: object definitions, relationships, and property sets:

- IfcObjectDefinition captures tangible object occurrences and types
- IfcRelationship captures relationships among objects
- IfcPropertyDefinition captures dynamically extensible properties about objects

A detailed illustration of the above mentioned instances can be accessed here[65]

Table 2-2: IFC Layers and their contents.

Layers	**Contents**
Resource Layer	Fundamental concepts expressed as entity types such as geometry (point, line and curve) topology (vertex, edge, face and shell), geometric model (CSG, B-Rep, Geometric Set).
Core Layer	Provides the basic structure of the IFC object model and defines most general concepts that will be specialized by higher layers of the IFC object model.

-The Kernel	Provides all the basic concepts required for IFC models within the scope of the current IFC Release. It also determines the model structure and decomposition. The Kernel can be seen as a template model that defines the form in which all other schema within the model is developed. Its constructs are very general and are not AEC/FM specific
-Core Extensions	Core Extensions, provide extension or specialization of concepts defined in the Kernel. More specifically, they extend those constructs for use within the AEC/FM industry. Each Core Extension is a specialization of classes defined in the Kernel and develops further specialization of classes rooted in the IfcKernel. Additionally, primary relationships and roles are also defined within the Core Extensions.
Interoperability Layer	This layer defines basic concepts for interoperability between different domain extensions. Shared building elements like beam, door, roof, window or ramp are defined in this layer.
Domain Layer	Domain Models provide further model detail within the scope requirements for an AEC/FM domain process or a type of application. Examples of Domain Models are Architecture, HVAC, FM, Structural Engineering *etc*. The primary IFC element hierarchy is based on the accessing structure, Project > Sites > Buildings > Stories > Spaces> Elements

2.6.1 IFC Schema and Contents

The IFC specification is written using the EXPRESS data definition language, defined as ISO10303-11 by the ISO TC184/SC4 committee. It has the advantage of being compact and well suited to include data validation rules within the data specification [66]. An example of the schema is shown in the Figure 2-2.

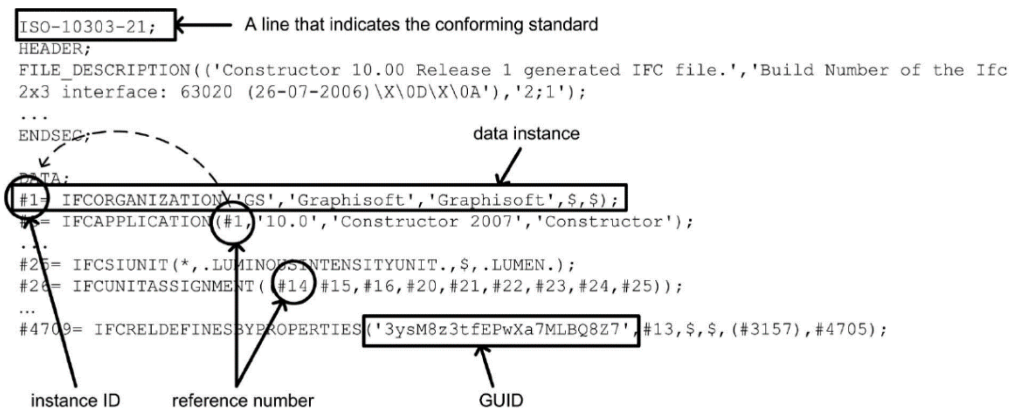


Figure 2-2: Part of an IFC file [67]

The IFC schema functions on a 'ladder principle' i.e. a lower layer class may not refer from a higher layer but can only refer from same or further lower layer class. Same layer referencing needs careful design and is only permissible within the Core layer and the Resource layer. ifcXML is an additional publication to the IFC-EXPRESS specification (since the IFC2x release), it has the XML document structure. ISO10303-28 ed. 2 defines the "XML representation of EXPRESS schemas and data" which is automatically created from the IFC-EXPRESS source. The data consistency for both IFC-EXPRESS and ifcXML is ensured by this standard. Consequently, it also enables the bi-direction conversion from *.ifc and *. ifcXML data files [68].

There is an anticipation that by offering an XML representation of IFC data, the application area will facilitate a broader community by providing access to a unified schema representing the built environment and related resources. XML has a broader range of supporting utilities and database implementations and is the basis for most e-Commerce messages and Web services. Another advantage of XML is that most web browsers and related application support this format, making the information immediately accessible on workstations and most other handheld computing devices. By observing the IFC content and semantic the XML community will benefit from an internationally accepted and agreed upon standard for the AEC-FM industry that has already been widely tested in several domains using other representations [69].

An IFC file can represent the following objects [68].

- Spaces, Space Structure (space, storey, part, building, site), Structural analysis (structural members, boundary conditions, connections, supports, loads, etc.), Zones / Compartments (fire, workstation, rising ducts, shafts)
- Shape (explicit), Shape (extrusions such as beams, pipes, ducts, walls etc.), Shape (topology such as line representations for pipe, duct, *etc*.)
- Structural Elements (members, profiles, rebars, properties, joints, features, surface)
- Building Elements (wall, door, window, roof, stairs, *etc*.), Relations Between Elements (holes, chases, voids, zones), Furniture (inc. system furniture), Sanitary Elements (WC's, urinals, baths, bidets, traps, gulleys)
- HVAC Equipment (chillers, fans, pumps, boilers, coils, cooling towers, heaters, heat exchangers, *etc*.), Electrical Elements (transformers, motors, generators, switches, protective devices, power and communication outlets panels, cubicles *etc*.), Fire Protection Elements (sprinklers, hose reels, hydrants, wet/dry rising mains)
- Systems (piping, ducting, cable, structural *etc*), Lighting (fittings, rendering, photo-accurate lighting), Grids, Manholes (manholes, inspection chambers, access chambers, meter chambers, valve chambers), draught
- Controls/Instruments (sensor, actuator, controller, gauge, meter), Draught, Time Series (time related events), Holes and Bases (holes, sleeves, packing,

framing, upstands, vibration isolation), Constraints (rules, specifications, requirements, trigger conditions), Accessories (brackets, drop rods, steel sections, bracket assemblies, screws, bolts *etc*.)

- Environmental Impact (embodied energy, CO_2), Asset Management (maintenance history, inventories), Help (request, action, permit, warranty, operation), Actors (people, organizations, addresses),
- Costing (cost planning, estimates, budgets, whole life), Work Plans and Schedules (including nested schedules, resource allocation) , Orders (work orders, change orders, purchase orders) Associated Documents Classification (OCCS, Uniclass *etc*.),
- External Data, Connectivity (services, structure, building), Geographical Elements features, contours, regions), Coordinate Mapping (geodetic, Cartesian).

2.6.2 Extension of IFC

Not all construction information is currently mapped on the IFC, therefore in order to reflect any additional information, an extension to the IFC must be carried out. Not much literature is available on the domain of extension of IFC. In a study conducted by Motamedi et. al [70, 71], an extension of IFC for Radio Frequency Identification (RFID) was proposed for facility management; which includes defining RFID system and establishing the relationship of the system with the other objects. These tags enable the users to access asset location and maintenance data.

The resulting IFC file is shown in the Figure 2-3 below. Property set assignment concept of IFC provides the basis for defining RFID systems' properties. Electrical Device Common, Condition, Environmental Impact Indicators, Manufacturer

(Type and Occurrence), Service Life, and Warranty are some available properties that are used. The material and casing of the tag is defined IFCMaterialUse, this is of significant importance since tag's radio communication capability is influenced by it when attached to metallic objects.

To further integrate this new entity with the rest of the model a set of relationship entities are introduced. IfcRelDecomposes and its subtype defines the decomposition relationship between an RFID tag and the associated building element. IfcRelAggregates are used to realize this relationship between tags and their associated elements. In order to describe the physical connectivity between an RFID tag/reader and a building element, IfcRelConnectsElements together with IfcConnectionGeometry are used. IfcConnectionGeometry is added to describe the geometric constraints of the physical connection of two objects [72].

EXPRESS code	Comment
/* Definitions */	
#139022=IFCBUILDINGELEMENTPROXY('*GUID*',#41,'Chiller_1',$,'148 Tons',#139021,#139012,'365499',.ELEMENT.);	Definition of the chiller "Chiller_1"
#289712=IFCBUILDINGELEMENTPROXY('*GUID*',#41,'LT1',$,'RFID Active Tag',#289711,#289706,'454896',.ELEMENT.);	Definition of active location tag "LT1"
#287118=IFCBUILDINGELEMENTPROXY('*GUID*',#41, 'AT_CH1',$,'RFID Passive Tag',#287117, #287112,'450749',.ELEMENT.);	Definition of passive tag "AT_CH1"
#384=IFCSPACE('*GUID*',#41,'1',$,$,#348,#382,'Room_1',.ELEMENT.,.INTERNAL.,$);	Definition of "Room_1"
#668=IFCSPACE('*GUID*',#41,'3',$,$,#606,#666,'Corridor_1',.ELEMENT.,.INTERNAL.,$);	Definition of "Corridor_1"
/* Coordinates */	
#286239=IFCCARTESIANPOINT((-7797.6,-19299.2,736.4));	Coordinates of passive tag "AT_CH1"
#289285=IFCCARTESIANPOINT((-8462.1,-18433.7,758.9));	Coordinates of active location tag "LT1"
#139014=IFCCARTESIANPOINT((-8176.7,-18055.8,50.9));	Coordinates of the chiller "Chiller_1"
/* Physical Relationships */	Attachment of Passive tag "AT_CH1"to the "Chiller_1":
#183985=IFCRELCONNECTSELEMENTS('*GUID*',#41,$,$,$,#139022,#287118)	Relationship (1)
/* Logical Relationships */	
#373558=IFCRELASSIGNSTOPRODUCT('*GUID*',# 41,$,$,#139022,#287118)	Assigning chiller to the passive tag: Relationship (2)
#373559=IFCRELASSIGNSTOPRODUCT('*GUID*',# 41,$,$,(#139022, #267844,#192183,#226763,#226855, #238888,#235,# 384),$,#289712)	Assigning assets and spaces to the active tag: Relationships (3), (4)
/* Spatial containment Relationships */	
#373318=IFCRELCONTAINEDINSPATIALSTRUCTURE('*GUID*',#41,$,$,(#107262,#109710,#109958,#111027, #111481,#111564,#111967,#112210,#139022,#159580,#159804,#159940,#160032,#192183, #194222,#244834,#267844,#289240,#291290),#384);	Containment relationship for assets inside "Room_1" including: Relationships (5), (6)
#373340=IFCRELCONTAINEDINSPATIALSTRUCTURE('*GUID*',#41,$,$,(#107890,#108403,#108556,#112859, #113354,#113837,#127697,#128030,#128131,#244264,#244345,#244636,#270365,#280369, #282636,#290391),#668);	Containment relationship for assets inside "Corridor_1" including: Relationship (7)
/* Properties Values */	
#373561=IFCPROPERTYSINGLEVALUE('AssessmentDate',$,IFCDATE('2014–02-02'),$);	Inspection date
#373562=IFCPROPERTYSINGLEVALUE('AssessmentCondition',$,IFCLABEL('Good-8/10'),$);	Inspection results (condition)
#373563=IFCPROPERTYSINGLEVALUE('EPCNumber',$,IFCIDENTIFIER('urn:epc:id: sgtin:0134000.213254.101'),$);	EPC number
/* Property Sets Definitions */	
#373571=IFCPROPERTYSET('*GUID*',# 41,'Pset_Condition',$,(#373561, #373562));	Condition property set
#373572=IFCPROPERTYSET('*GUID*',# 41,'Pset_RFIDSystemPassiveTag',$,(#373563, #373564, #373565));	Passive RFID property set
/* Relating Property sets to elements */	
#373573=IFCRELDEFINESBYPROPERTIES('*GUID*',# 41,$,$,(#139022), #373571);	Relating condition property set to "Chiller_1"
#373574=IFCRELDEFINESBYPROPERTIES('*GUID*',# 41,$,$,(#287118), #373572);	Relating RFID property sets to passive tag "AT_CH1"

Figure 2-3: Part of the Express Code for the model [72]

2.6.3 Cost Estimation through IFC

The previous studies mentioned have ignored the importance of retaining the LCA results within the BIM environment; the results are in the form of excel sheets or pdfs. This is probably because there is not adequate mapping in the underlying IFC schema which can hold these values and make them available for automated decision-making systems. On the other hand, *Cost* is adequately mapped in the schema and thus it is represented in an orderly fashion. It is vital to examine the embedment of *Cost* in the IFC model before proposing IFC extension for energy.

The estimation of energy is in many ways similar to the estimation of *Cost* as both can be assigned to an individual material, element, equipment use, transport haul etc. Therefore, it is imperative to study the application of cost estimation within IFC. An IFC-based construction product information model of Tender of Building Project (TBP) was developed by Ma [73] which is used for the development for cost estimation for the architectural and structural engineering of the cast in place concrete (CIPC) structure projects by using relevant entities and relationship entities from the IFC standard, as shown in Figure 2-4.

The decomposition of building elements into construction products is expressed Fig. 2-5 (part A). Three entities namely IfcCovering, IfcReinforcingElement and IfcBuildingElementPart derived from IfcElement represent construction products such as decorations, reinforcement members, and other subordinate substances of building elements such as concrete core layer. IfcRelAggregates and IfcRelCoversBldgElements act as relationship entities representing decomposition relationships between decomposing construction products and its building element.

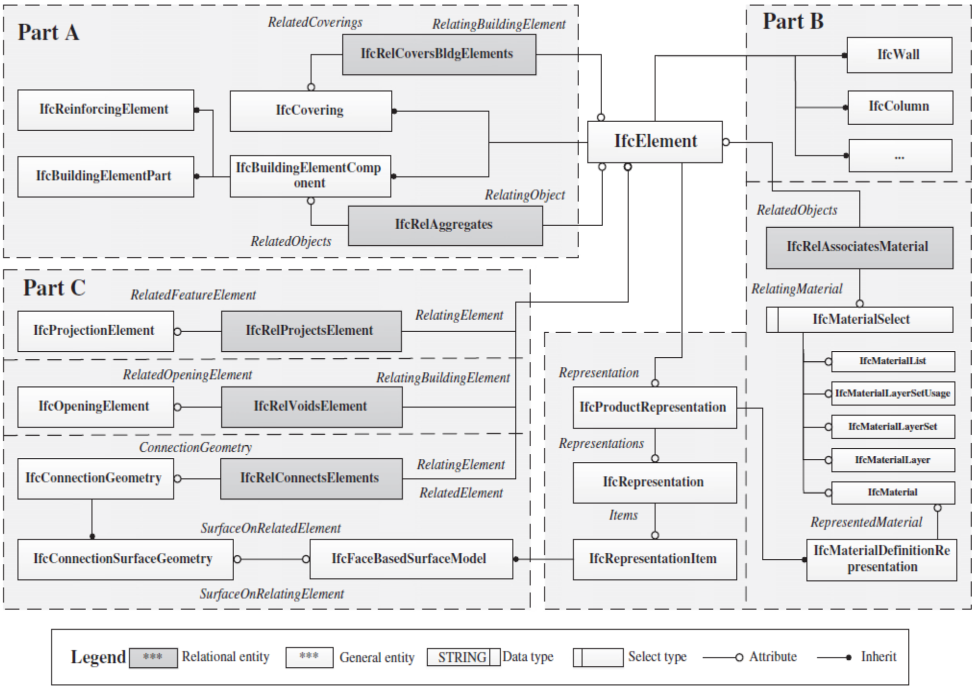


Figure 2-4: IFC-based construction product information model of TBP cost estimation [73]

Three types of information i.e. geometric information, material information, and product type are expressed in Part B of Fig. 2-4 that can be transmitted to construction products from building elements in the design model for TBP cost estimation. IfcWall and IfcColumn are the subclasses of IfcElement which expresses the Product type. IfcProductRepresentation and other relevant entities (including IfcRepresentationItem whose subclasses such as SurfaceModel, Brep and CSG) expresses the geometric information. IfcMaterialSelect expresses the material information by acting as select type among IfcMaterialLayerSetUsage, IfcMaterialList, IfcMaterial and IfcMaterial Layer [73]. Intersection relationships among construction products as well as building elements, opening and accessories are expressed in Part C of Fig. 2. IfcRelConnectsElements, IfcRelProjectingElement and IfcRelVoidsElement express three relationships respectively. For the intersection part geometric information is expressed by IfcFaceBasedSurfaceModel and IfcConnectionSurface [73].

A broad map for IFC based; specification compliant; semiautomatic TBP cost estimation is shown in Figure 2-5. There are four main processes *i.e.* automatic decomposition into construction products, semi-automatic classification of construction products into cost items, automatic quantity takeoff for cost items, and automatic project cost computing [73].

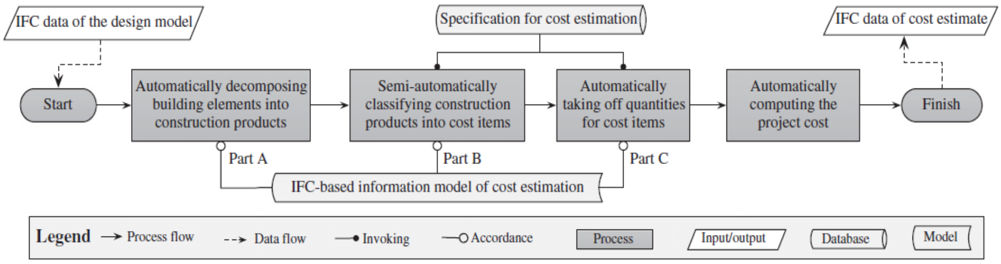


Figure 2-5: General process map and key algorithms for semi-automatic and specification-compliant TBP cost estimation [73]

2.6.4 Analyzing Cost Embedment in the IFC Model

After reviewing an application of the cost estimation using IFC, the integration of *Cost* information in the IFC model is a key step. Using a bottom-up approach to analyze the IFC architecture [74], primarily yields an independent *Ifccostresource* which provides the basic schema to identify cost values. This *IfcCostValue* may be an absolute value or a multiplying factor for some other cost value. Usually this value is per unit quantity basis to be consistent with the prices available per unit quantity of an item. In this manner, the total cost can be calculated by combining a set of component values. This is achieved through the assertion of an applied value (*IfcAppliedValue*) relationship which acts as a container for applied value components. Apart from other text or numeric attributes, *IfcAppliedValue* class has two select attributes: *IfcAppliedValueSelect* and *IfcArithmeticOpertorEnum*. The

IfcAppliedValueSelect constitutes of a combination of attributes while *IfcArithmeticOpertorEnum* is an enumeration of simple arithmetic operations such as add, divide, multiply and subtract. Another attribute of *IfcCostResource* is *IfcCurrencyRelationship* which deals with the exchange rate of currencies.

Moving up the hierarchal ladder of the IFC model, just above the Resource layer is the Core Layer. One of the schemas of the core layer is the *IfcKernel*, which defines the most abstract part or core part of the specification. It captures general constructs, that are basically founded by their different semantic meaning in common understanding of an object model, like object, property and relationship. One of the object models of *IfcKernel* is *IfcControl*, which is the abstract generalization of all concepts that control or constrain the utilization of products, processes, or resources in general.

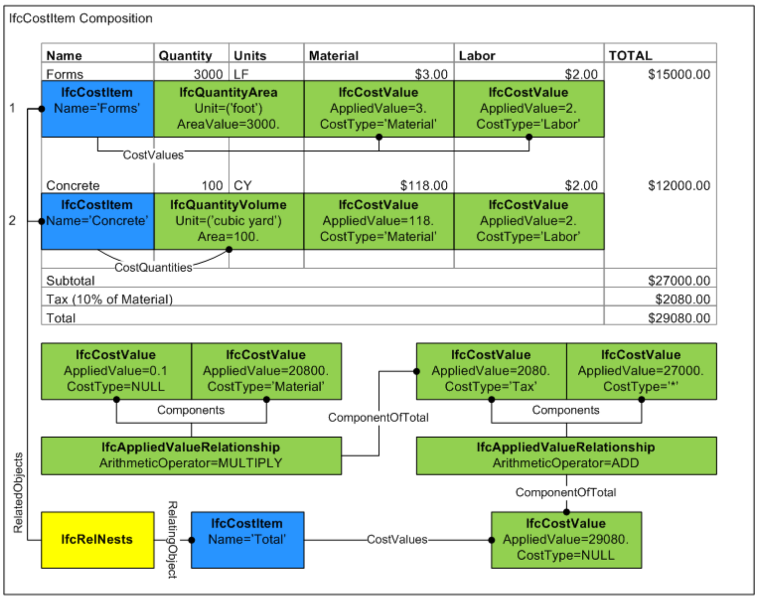


Figure 2-6: *IfcCostItem* Cost Composition [75]

Two cost related instances are within the subset of *IfcConrol*: *IfcCostItem* and *IfcCostSchedule*. However, these entities lie within the shared management elements of the Interoperability Layer of the IFC architecture. The *IfcSharedMgmtElements* schema defines basic concepts that are common to management throughout the various stages of the building lifecycle. *IfcCostItem* is used for cost estimates, budgets, and other forms, where a variety of identification codes are used extensively to identify the meaning of the cost. To further clarify the context of *ifcCostItem*, an example is illustrated in the Figure 2-6.

The previously discussed concept of *IfcCostValue* is used as a datatype for the value of *IfcCostItem*. *IfcRelNests* relationship is used to sum up the difference instances of *IfcCostItem* and retain the value in "Total" instance of the *IfcCostItem.* Moving on to the other cost related instance of *IfcControl, an IfcCostSchedule* consolidates the instances of *IfcCostItem* to generate estimates of constructions costs for various purposes. Apart from these entities, two cost related types also exist in the *IfcSharedMgmtElements.* These types are *IfcCostItemTypeEnum* (user defined or undefined type) and *IfcCostScheduleTypeEnum* (Budget, Tender, BOQ etc.).

The top most layer of IFC model, the domain layer constitutes of different domains such as Architecture, HVAC, Building Controls etc. The *IfcConstructionMgmtDomain* accounts for the costs incurred on the project due to different resources and these resources are represented by *IfcConstructionResource,* which is an abstract generalization of the different resources used in construction projects, mainly labor, material, equipment and product resources, plus subcontracted resources and aggregations such as a crew resource. *IfcConstructionResourceType* further determine the specialty of resource and provide classification related to the tasks it would be needed for.

Before summarizing the discussion of cost embedment in the IFC architecture, it is vital to elaborate another resource known as *IfcQuantityResource*. *IfcQuantityResource* defines a set of basic quantities that can be associated with products through the *IfcElementQuantity* (defined in *IfcProductExtension*). This *IfcElementQuanitiy* contains physical quantities such as Count, Length, Area, Volume, Time etc. To emphasize the importance of this resource, a diagram for cost assignment on elements is shown in the Figure 2-7.

To summarize the role of various layers and different components within these layers in embedding the cost information in the IFC architecture, a top down approach is employed. Table 2-3 shows the different layers of the IFC structure with the cost relevant components and the related IFC entities used to embed the cost information in the current IFC4 model.

Table 2-3: Summary of IFC cost embedment

IFC Layer	IFC Layer component	IFC Entity and Types
Domain Layer	*IfcConstructionMgmtDomain*	*IfcConstructionResource* *IfcConstructionResourceType*
Interoperability Layer	*IfcSharedMgmtElements*	*IfcCostSchedule* *IfcCostItem* *IfcCostItemTypeEnum* *IfcCostScheduleTypeEnum*
Core Layer	*IfcKernel*	*IfcControl*
Resource Layer	*IfcCostResource*	*IfcCostValue* *IfcCurrencyRelationship* *IfcArithmeticOperatorEnum* *IfcAppliedValueSelect*

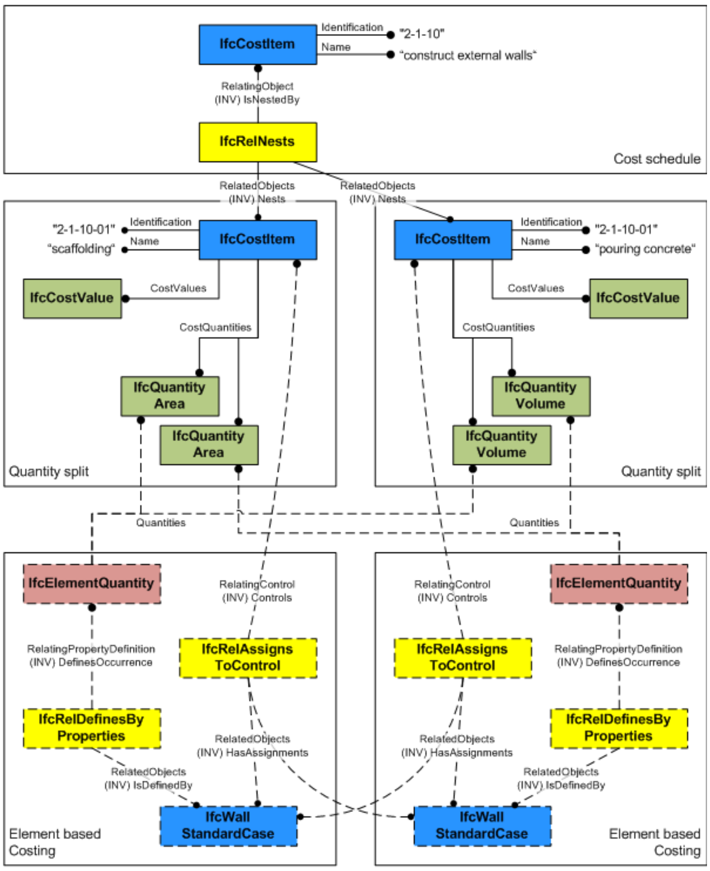


Figure 2-7: IFC Cost Assignment [76]

2.6.5 Embodied Energy in the IFC Model

In the current IFC4 model, the embodied energy information is embedded as a property (*TotalPrimaryEnergyConsumption*) within a property set of *Pset_EnvironmentalImpactValues* lying in the core schema of *IfcProductExtension*. *IfcEnergyMeasure* is lowest level representation shared in the *IfcMeasureResource* of the Resource Layer. To put things into perspective, the other instances of *IfcMeasureResources* are *IfcForceMeasure*, *IfcAreaMeasure etc.* which are purely physical properties with no prospect of optimization or decision- making inputs in the larger context of a construction project. Interestingly, within the

IfcMeasureResource there is a type of *IfcMonetaryMeasure*, which again is just a value with no relation with the above-mentioned concept of cost embedment.

2.7 Summary

A literature review around six domains; Material Energy, Transportation Energy, Construction Energy, BIM, Construction Simulation and IFC is conducted. Some salient findings of the literature review are listed below:

- Construction energy is neglected in most of the studies
- Most of the studies claiming to use BIM, only used some BIM focused quantity take-off tool and exported the quantities to some other platform
- Limited literature focused on the simulation and optimization of embodied energy was found
- Sustainability information is not mapped to BIM data structure in any of the studies till date.

CHAPTER 3: METHODOLOGY

The method in this research consists of a proposed framework and system prototype development. The proposed framework illustrates the theoretical divisions, mandatory procedures and the data exchange for the assessment of embodied energy associated with the selection of building materials, construction equipment and inclusive transportation along with the installation of building assembly on site. The developed prototype refers to the design and development of the tool that implements the framework and serves as a platform for testing the applicability of the framework in the case study of this research. The discussion about the system prototype development is deferred to Chapter 4 of this thesis.

The scope of the current research is limited to the estimation of initial embodied energy. Figure 3-1 shows the proposed framework which comprises of five steps; data collection, data mapping, data analysis, simulation models and IFC embedment.

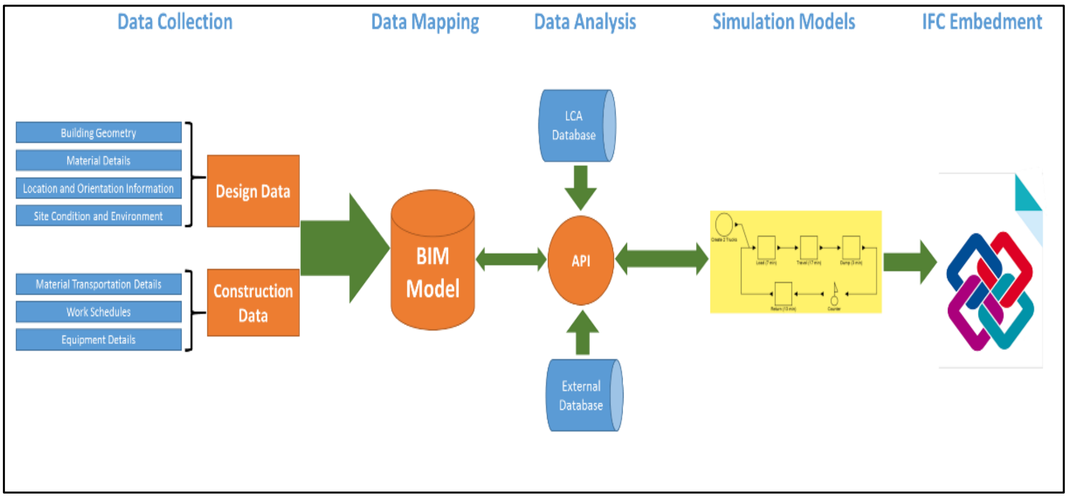


Figure 3-1: Research Methodology

3.1 Data Collection

In order to evaluate the embodied energy, all the direct and indirect processes associated with a building before the operation stage should be taken into consideration, including energy consumed during material extraction, manufacturing, transportation and construction. Therefore, data needs to be collected in a systematic manner such that none of the mentioned aspects is ignored. It is also important to understand the sources of data in detail, because based on these sources the data collected is divided into two categories. The first category includes data that is associated with the design and construction of a particular project (further referenced as 'project data') and the other category includes external data that is common for all the projects executed in that vicinity (further referenced as 'common data').

The project data encompasses the geometry of the buildings, the material used in the building elements, location and orientation of the building, location of manufacturing plants of materials, construction methods, work schedules etc. The common data in our case is the embodied energy coefficients of the construction materials (extracted from the ICE database), transportation coefficients as well as energy coefficients for equipment which are extracted from the Chinese construction handbook [77].

As mentioned previously, the ICE database [13] is a leading embodied energy and carbon database for building materials which is used in this research. It contains data for over 200 materials, broken down into over 30 main material categories. The "cradle to gate" approach was used to calculate the embodied energy in the database, which considers all energy consumption from upstream stages such as raw material extraction to the final stage as a finished product. Table 3-1 shows some materials popularly used in construction.

Table 3-1: Embodied energy coefficient for typical construction materials [13].

Material	**Embodied Energy & Carbon Coefficients**		
	EE – MJ/kg	**EC – kgCO2/kg**	**EC – kgCO2e/kg**
Aggregate	0.083	0.0048	0.0052
Aluminum	155	8.24	9.16
Cement	4.5	0.73	0.74
Concrete	0.75	0.1	1.107
Iron	25	1.91	2.03
Sand	0.081	0.0048	0.0051
Steel	20.1	1.37	1.46

In order to further investigate the project data, a questionnaire was designed to collect this data. This questionnaire was then shared with construction site manager. In response to the questionnaire a set of documents were recived from the construction staff and the questionnaire was filled with help of those documents. The questionnaire mainly spanned over five parts, since during the time of the survey excavation was underway, the first part covered the data collection about the excavation and earth moving activity. Details such as zoning for excavation as well as the start finish times were asked. In the next part data was collected about different material manufactories, warehouses and machine leasing companies with respect to their locations and preferred routes for the estimation of transportation distances.

In the third part, data was collected for the supporting activities such as formwork and scaffolding. In the fourth part, information about the transportation vehicles was collected; the prime focus was on the fuel consumption of different types of trucks. In the fifth part, data was collected for the transportation of staff and heavy equipment. A detail of the collected data is provided in section 6.1. The questionnaire is attached as appendix A.

3.2 Data Mapping

A typical BIM authoring tool has provisions for assigning geometry and location data, as well as material properties, but it lacks provisions for embodied energy of materials, transportation energy and construction energy. Therefore, as per the categorization provided in the data collection step, common BIM authoring tools partially capture the project data but provide inadequate support for the common data. Hence, further development on top of the BIM authoring tool is necessary to embed the common data in the BIM model. This development, for instance, includes matching of embodied energy coefficients with the BIM model materials and extraction of the corresponding values from the LCA database. In its entirety, data mapping is responsible for linking external data with the BIM environment to estimate the embodied energy.

Three kinds of external data are required to calculate the energy consumption: embodied energy coefficient of materials, material transportation vehicle types with fuel efficiencies and construction equipment efficiencies data. For material embodied energy coefficients, ICE database was imported in the Structured Query Language (SQL) server. And for the transportation and construction energy coefficients, the data collected during survey for material transportation and construction equipment efficiency was tabulated in SQL Server in the form of a database.

Once these external databases were made, a plugin was developed to link these external databases with the BIM model and perform calculations. The plugin is supposed to intelligently match the external database tables with the BIM data to evaluate embodied energy values (detail is presented in chapter 4). Figure 3-2 shows the proposed data flow and integration for evaluating embodied energy which was used to develop the plugin. The plugin is basically linked to two types of databases; one is the external database which is already discussed above and the second is BIM database which exists in every BIM authoring tool and holds the BIM model.

The data retrieving algorithm in the plug in was developed to query data from both these databases. SQL is used to retrieve the required data from the database and the plug-in itself is developed in C#. At the front end, a user interface (UI) is developed on top of the BIM authoring tool as a part of the plug-in to display the data retrieving options and their respective results. The whole process is pivoted on the research aspect which includes shortlisting of embodied energy coefficient database, designing survey tool and developing BIM model.

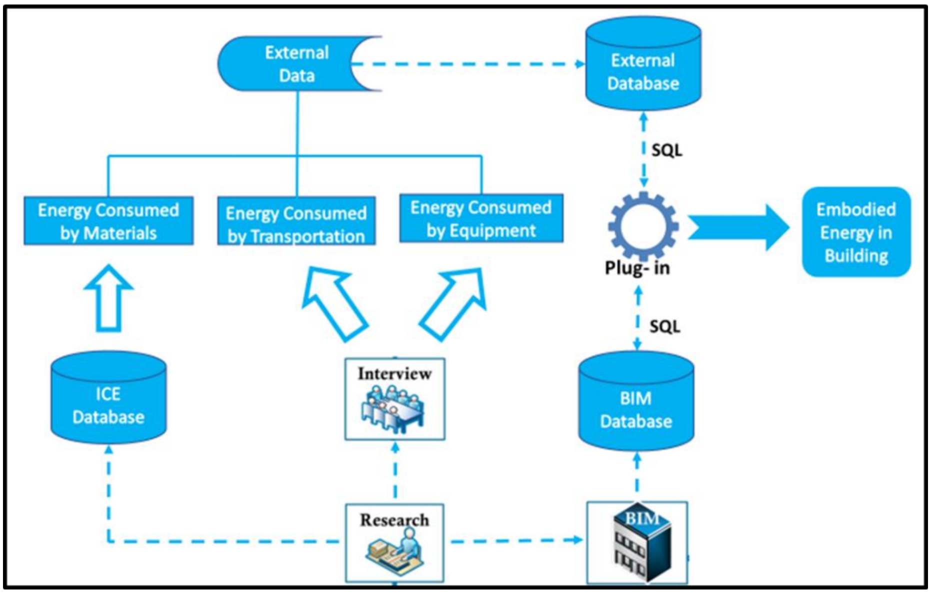


Figure 3-2: Data flow and integration

3.3 Data Analysis

This step includes determining the total embodied energy (EE_T), which is the sum of material embodied energy (EE_M), material transportation energy (EE_{TR}), and construction energy (EEC). The total embodied energy, EE_T of the building can be computed by using equation 3.1.

$$EE_T = EE_M + EE_{TR} + EE_C \tag{3.1}$$

Material embodied energy is calculated by summing up the material embodied energy of all the materials used in project as shown in equation 3.2.

$$EE_M = \sum_i^n EE_{m_i} \tag{3.2}$$

Material embodied energy for individual material (EE_m) can be calculated by multiplying the quantity (volume) of the material to its density (p_m) and embodied energy coefficient (ec_m). The quantities available from the BIM authoring tool are in volume unit, which must be multiplied by its density to get the mass which can be used with embodied energy coefficients. It is important to note the material quantity should be extracted from all the elements using that material and summed up to get the total quantity of a material. Density can be obtained from the properties of the material modelled in the BIM software. The embodied energy coefficient will be obtained from the ICE database. Equation 3.3 illustrates the computation for individual material embodied energy.

$$EE_m = \sum_{i}^{n} q_i p_m \, ec_m \tag{3.3}$$

Material transportation energy (EE_{TR}) is calculated by summing up the material transportation energy of all the materials used in the project as shown in equation 3.4.

$$EE_{TR} = \sum_{i}^{n} EE_{tr_i} \tag{3.4}$$

Material embodied energy for individual material (EE_{tr}) can be calculated by linking hauling distance of material (D_m), round trips (R_{tm}), fuel energy coefficient (F_{Vec}) and lower heating value of fuel (Lv_F). Equation 3.5 illustrates the computation for individual material transportation energy (EE_{tr}).

$$EE_{tr} = R_{tm} D_m F_{Vec} Lv_F \tag{3.5}$$

The construction energy can be calculated by determining the equipment energy consumed for performing different work packages. The standard work packages are established using the logical break down structure of a building project including foundation works, columns, beams, floors, walls and roofs. The equipment used for performing a work package is obtained by referring to different construction practice manuals and handbooks [77]. Equipment productivity and power ratings are obtained from similar handbooks. These values are left editable in the interface to change as deemed necessary. A separate database is made to retain the equipment related data so that further inclusion or modification of this data is possible and retrievable at a later stage. Equation 3.6 illustrates the computations for construction energy (EE_C).

$$EE_C = \sum_{i}^{n} EE_{CWP_i} \tag{3.6}$$

The construction energy for individual work packages (EE_{CWP}) can be calculated by summing up the energy used by different equipment to perform the work package as shown in equation 3.7.

$$EE_{CWP} = \sum_{i}^{n} ee_i \tag{3.7}$$

The energy consumed by each piece of equipment (ee) to perform a work package can be calculated by multiplying rated power (R_e) of the equipment with the estimated working hours (Wh_e) of the equipment as shown in equation 3.8. A conversation factor of 3.6 is used to convert from Kilo-Watt hour (KW.h) to Mega-Joule (MJ).

$$ee = 3.6\, Rp_e Wh_e \tag{3.8}$$

The working hours of the equipment are calculated by using the daily productivity of the equipment and the quantity of work to be performed.

3.4 Simulation model development

Construction process simulation models are developed to analyse the repetitive activities. These repetitive activities can be a whole construction activity itself such as concreting of twenty floors or part of a construction activity such as one roundtrip of a truck in the context of an earth moving operation. Figure 3-3 illustrates a schematic of an earth moving operation [45]. There are so many parameters in this activity that can significantly impact the durations as well the energy consumption of the activity and the interrelation between these parameters are quite complex.

For instance, a general perception can be that by increasing the number of hauling trucks, duration of the activity may reduce. However, this will happen to a certain extent where the utilization of the loader resource does not reach the maximum. Similarly, the size of the truck also plays a vital role specially in the context of energy consumption. Therefore, the optimum solution would be to find a combination that can provide reasonable duration and energy consumption.

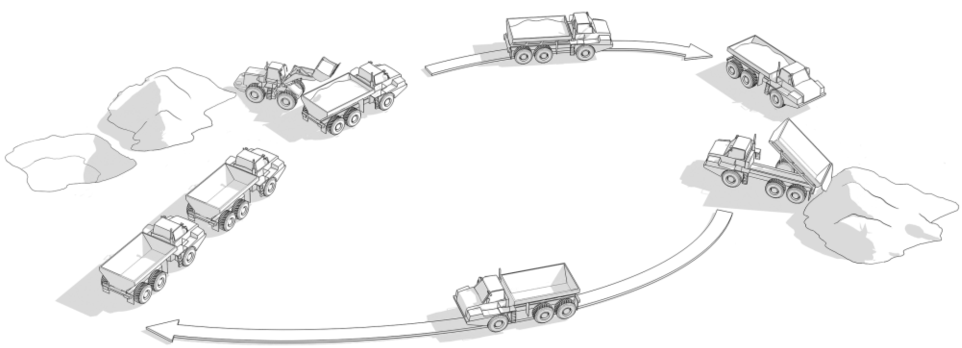


Figure 3-3: Schematic of an earth moving operation [45]

It is important to note that the above-mentioned example was just an earth moving example, now if the excavation also becomes a part of it, the efficiency of the excavators also becomes a factor. Almost all the activities in construction are respective and can be represented as a simulation model such as concreting, brick laying, flooring, finishes etc. And by changing the quantities and sizes of construction resources involved in each activity, optimized combination can be identified in terms of duration and energy consumption.

To automate the current simulation practices with the help of BIM is the prime concern. To achieve this automation, efforts needs to be focused in two domains; integration of simulation engines in the BIM environment and availability of pertinent data. This methodology focuses on the data availability for simulations rather than the inclusion of simulation engines which may limit the type and nature of simulations performed. Quantities of different building materials used in the project are obtained from the BIM model, which are combined with construction equipment and hauling vehicle productivities to obtain the duration of different activities.

The productivities of these machines under different circumstances are retrieved from the external database and combined with the fuel efficiencies and heating value coefficients to determine the energy consumption patterns of different activities. Once the energy consumption values are estimated, certain modifications regarding the number of available resources are performed based on expert judgment and the estimations are conducted for these newly created scenarios. By comparing various scenarios, a clear understanding of the underlying patterns regarding the interrelation of various resources and constraints can be obtained. This helps in making informed decisions with respect to the feasible and optimal number of resources for various project goals.

3.5 IFC Embedment of EE Data

The embedment of embodied energy data in IFC format is a critical part of this study as it makes this data interoperable with all the authentic BIM authoring tools. To perform such a feat, multiple tasks including deep exploration of the literature, hands-on data performance tests, identification of the shortcomings of the original IFC model, methodology development for alteration of the original model and implementation of the methodology are performed. The following Figure 3-4 presents the steps performed to embed the energy data in the IFC file format.

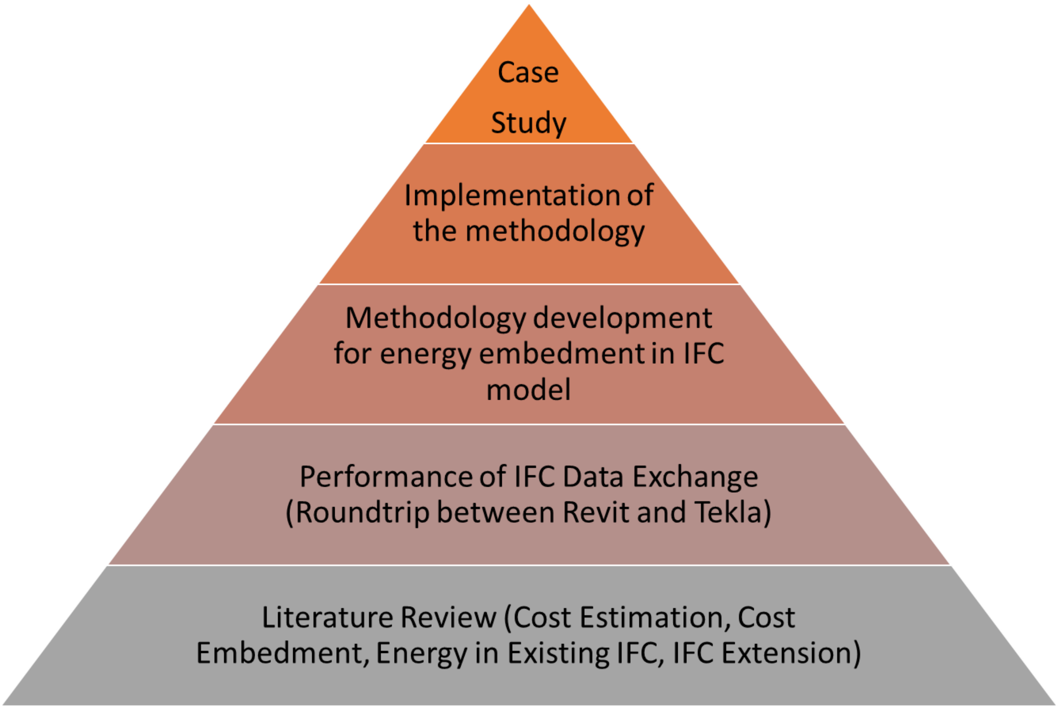


Figure 3-4: Method for energy embedment in IFC

3.5.1 Performance of IFC Data Exchange

In order to understand the interoperability function of the IFC data model as well as to assess the performance of this function a data exchange exercise is conducted. This exercise brings forth a data exchange analysis between two of the most popular BIM software to date: Revit 2016 and Tekla 2016. Two models (one story and two story) are modelled on Revit. Figure 3-5 shows the methodology to conduct this research exercise. Since both the software have their own IFC import-export method; the IFC files are imported and exported using their native tools. An alternate method of data exchange is also adopted using the additional feature of Tekla BIM sight which is able to import an IFC file and save as Tekla BIMsight project file (*.TBP) which can be imported into Tekla Structures as the reference model.

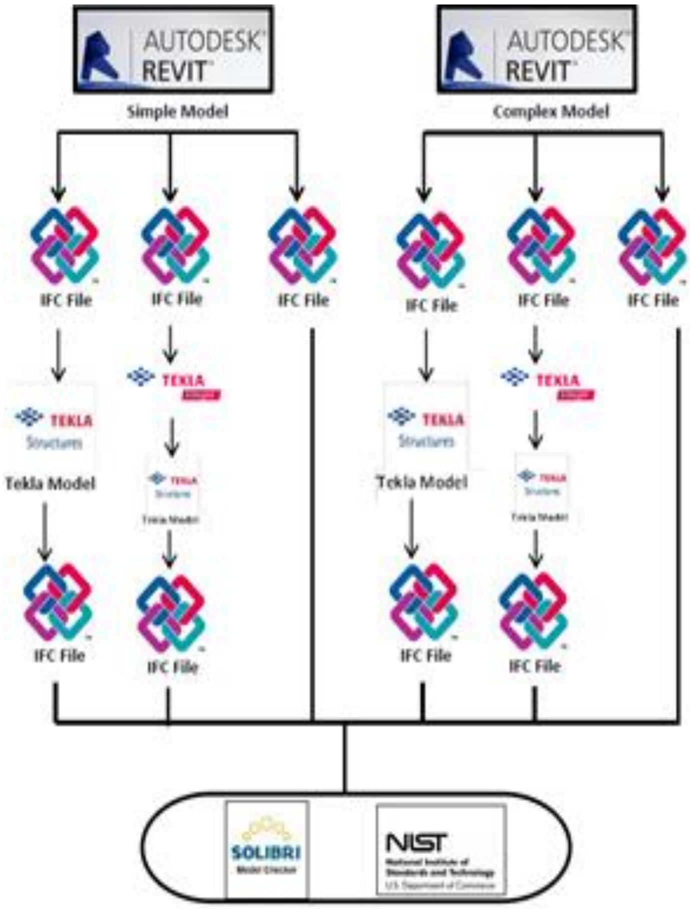


Figure 3-5: Methodology Flow Chart for IFC data exchange exercise

These files are then viewed and analyzed in the IFC checkers and analyzers like Solibiri Model checker and IFC Analyzer. The IFC files produced for both the buildings are compared interms of : (1) physical file-size; (2) differing numbers of instances; (3) inconsistent object types; (4) inconsistent attribute values (missing or new values, loss of numerical precision, string length differences, value differences, reference number differences, etc.); and (5) schema inconsistencies.

The software tools used to undertake this exercise are basically of three types. The first type includes the BIM authoring tools which are used to produce the design and influence on the instance modeling of the buildings such Revit and Tekla. The second type includes the export-import converters such as TEKLA BIM sight and the third type includes the analyzers such as Solibiri Model Checker (SMC) and IFC Analyzer which analyzes the data exchange between the BIM authoring tools. SMC is one of the most widely used checking applications [78]. SMC is a powerful checking platform that can also be used to perform e.g. clash detection and code compliance checking.

3.5.2 Methodology development for Energy Embedment in the IFC model

As discussed in the literature review section that cost embedment in the IFC model was found closest to the intention of energy embedment, so a methodology is developed to incorporate energy data in the IFC model by understanding cost embedment.

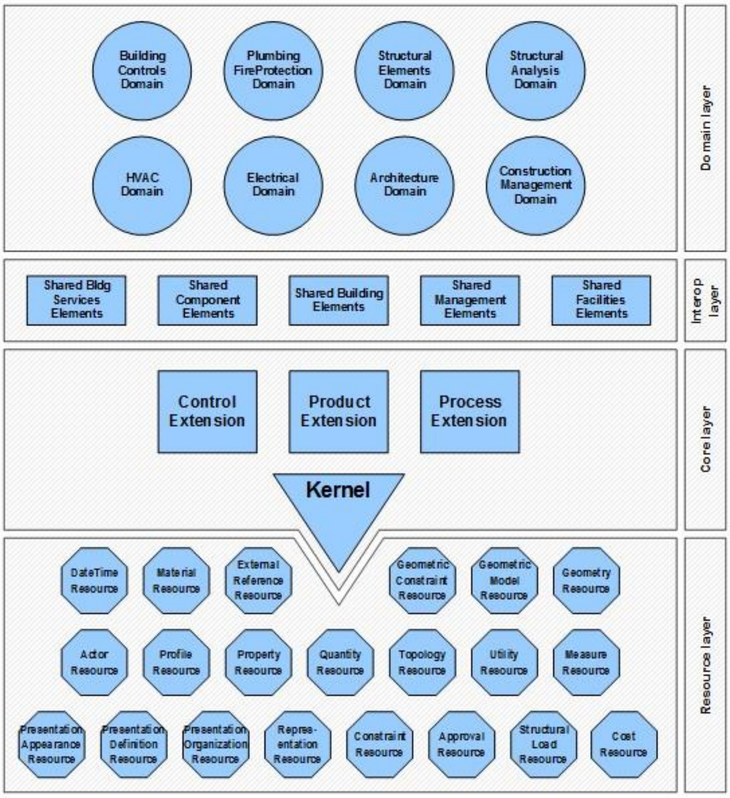


Figure 3-6: IFC data architecture schema with conceptual layers [79]

Within the IFC4 model, the most external and loosely bonded reference of cost was located and then its lead was followed to understand the weaving of cost model into the IFC architecture. Once the connection of the entities underlying all the layers was established, a table was generated to list all the entities and types with respect to its layer and corresponding layer component. Figure 3-6 above shows the IFC data architecture schema with conceptual layers.

Once the embedment of cost model was extracted from the IFC Model, the cost related entities and types were adopted and modified wherever necessary to propose new entities for embodied energy. Since, the availability of cost information is usually from an external source such as a pricelist of materials and products, and the availability of embodied energy information is also from an external source such a Life Cycle Inventory (LCI) database pivoted on material or product information, a stark similarity is encountered in the way both these models work. Therefore, with minimal changes new entities and types were proposed for the IFC extension and the existing entities fulfilling the purpose of both the models (cost and energy) were left unchanged.

3.5.3 Implementation of the methodology

Once all the proposed changes were identified, the implementation of these changes are performed by modifying an open source toolkit for building IFC based applications. The rationale behind using an authentic open source toolkit for IFC is to be able to modify the IFC files in such a way that they remain consistent with the IFC based applications. In this step several new classes were developed to match the additional entities required to sufficiently represent the embodied energy in the IFC model as identified in the previous step. After modifying the xBIM [80] tool to incorporate energy data, the prototype tool is enhanced to use the modified xBIM model to create IFC files with energy data. This step requires two sub task; the first included exporting IFC file from the BIM authoring tool and the second required modifying the IFC file to incorporate energy data.

3.6 Summary

The literature review conducted in chapter 2 highlights four broad limitations of the existing tools. The proposed methodology therefore aims to overcomes these limitations. The first limitation was that the studies were project based, which specifically means that the tools/methods developed were specially designed to cater certain projects and cannot be generalized. However, the current study is not designed keeping in view any specific project, rather it is designed to be flexible enough to fit any project.

The second limitation was that the studies were treating BIM as just a quantity takeoff tool. In contrast the current study keeps BIM at its centre, and the complete analysis as well as results reside within the BIM environment. The third limitation highlights the ignorance of construction phase. In response, the current study completely takes in the construction phase by incorporating construction equipment energy consumption and working durations. The final limitation of the previous studies points out the complexity and rigor involved to perform the embodied energy estimations. The current study simplifies and automates the assessment by allowing data integration from the BIM model as well as the external databases. And finally allows the export of the analysis results to IFC format to allow further data processing.

CHAPTER 4: PROTOTYPE SYSTEM DEVELOPMENT

To implement the proposed framework a prototype system is designed and developed. Autodesk Revit 2016 is used as the BIM authoring tool in this study and all the project data are assumed to be captured within the Revit environment prior to the functioning of the tool. The main development for the plug-in is performed using C# language in the Microsoft Visual Studio environment. The plug-in is based on the Standard Development Kit (SDK) for Revit 2016 provided by the Autodesk. The plug-in uses Application Programming Interface (API) of the Revit 2016 to become a part of Revit and access internal data protocols of the software. Querying or updating data in the external databases is mostly carried out through the SQL commands integrated in the main C# program.

4.1 Workflow for Material, Transportation and Construction Energy

The prototype tool comprises of three sections: Material Energy, Transportation Energy and Construction Energy. The material embodied energy section involves assessment of material embodied energy content in the building. This assessment requires two types of data: material quantities and embodied energy coefficient for materials. Material quantities can be extracted from Revit and the coefficients can be retrieved from the ICE database. Figure 4-1 illustrates the workflow of the material embodied energy section.

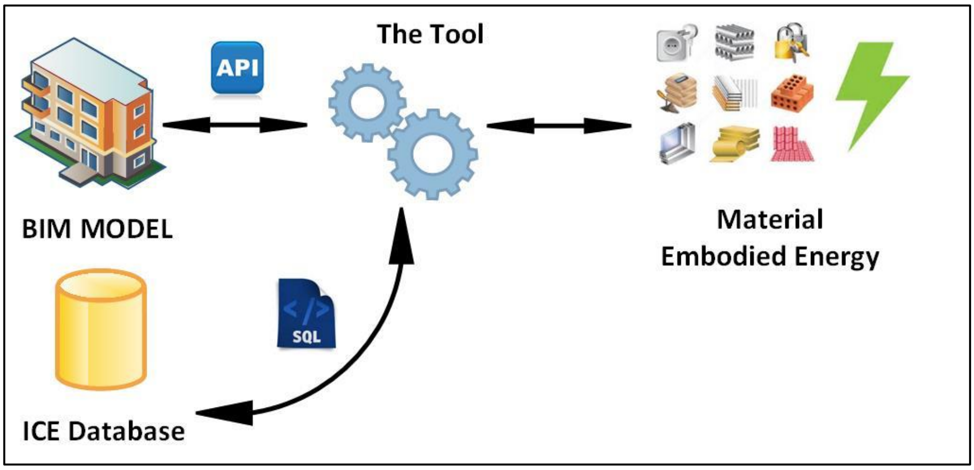


Figure 4-1: Workflow for material embodied energy assessment

The transportation energy assessment requires material quantities, vehicle types and capacities, and distances from supplier to sites. Material quantities can be extracted from the BIM authoring tool and the latter two can be selected from the designated databases. Figure 4-2 shows the workflow for the assessment of transportation energy.

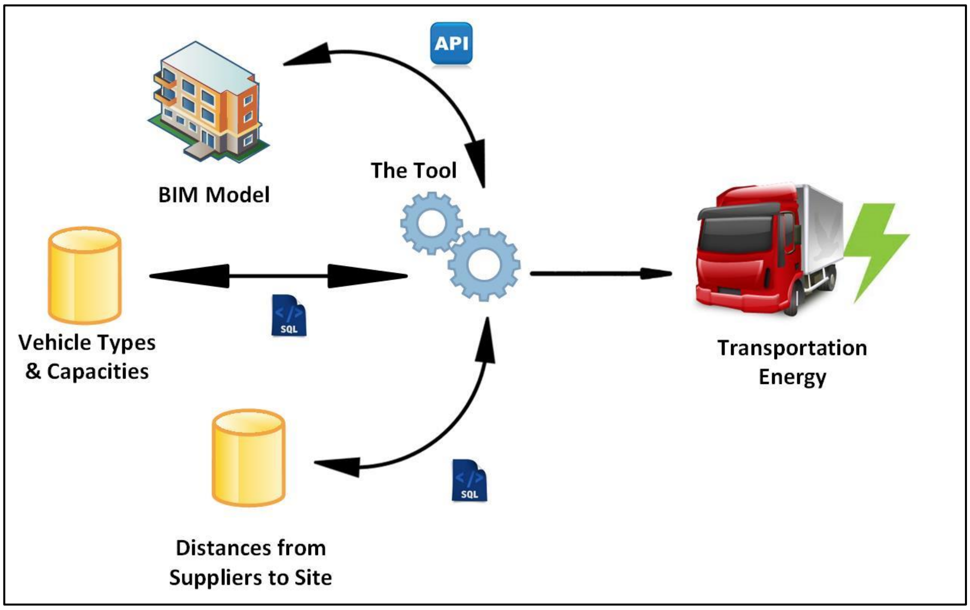


Figure 4-2: Workflow for transportation energy assessment

The construction energy assessment again requires material quantities, construction equipment productivities and power ratings, and work package information. Material quantities, as mentioned previously, can be extracted from the Revit, construction equipment productivities and power ratings can be obtained from the designated databases and standard work package information which is inbuilt in the tool. Figure 4-3 displays the workflow for the assessment of construction energy.

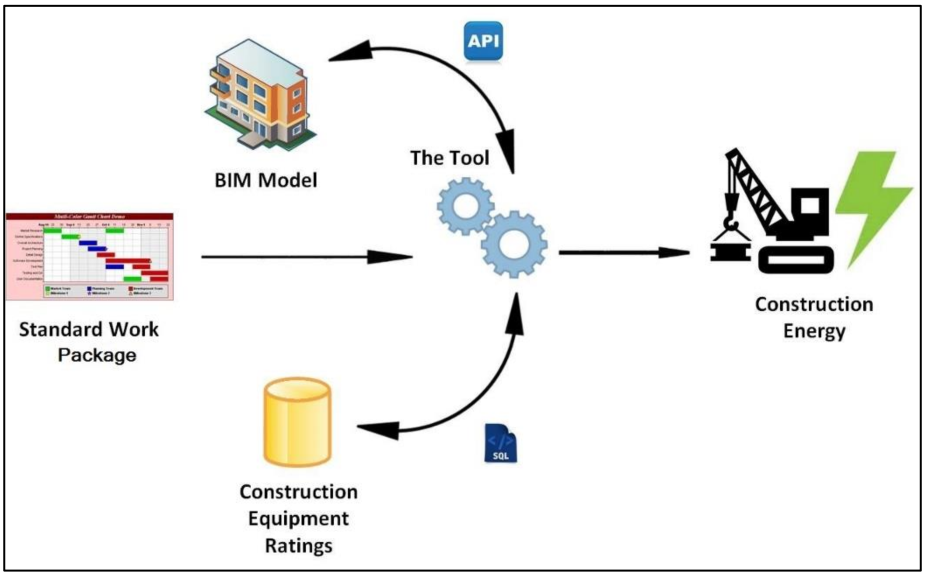


Figure 4-3: Workflow for construction energy assessment

4.2 Rationale for Prototype Development

As mentioned previously, the data needed to determine the embodied energy content is categorized into two types: project data and common data. Since the project data are partially captured in BIM environment, the real challenge is to realize the common data in the BIM environment.

This development aims to achieve four purposes:

1. Enable accessibility to the BIM model data: Revit provides a rich and powerful .NET API which can be used to automate repetitive tasks, extend

the core functionality of Revit in simulation, conceptual design, construction and building management, etc. Revit .NET API allows programming with any .NET compliant language including C# [81]. This accessibility is pivotal to the use of the building information stored in the model for embodied energy estimation, particularly for retrieving material information such as material names, properties and quantities.

2. Query data from the External databases (Ms Access) using SQL: SQL is a computer language for working with sets of facts and the relationships between them. Relational database programs, such as Microsoft Office Access, use SQL to work with data. Like many computer languages, SQL is an international standard that is recognized by standards bodies such as ISO and ANSI. The SQL queries embedded in the C# API are used to retrieve information stored in external databases outside the model, such as embodied energy coefficients from the ICE database or material hauling distances mentioned in the transportation database.

3. Provide a user interface for the estimation of embodied energy: This is accomplished by using Windows Forms. Windows Forms is a smart client technology for the .NET Framework, a set of managed libraries that simplify common application tasks such as reading and writing to the file system. When a development environment like Visual Studio is used, Windows Forms smart-client applications that display information can be created, which can also request input from users, and make that data available for further processing [82]. This user-interface not only displays all the considerations while estimating the embodied energy, but also shows the final results of the proposed approach. This acts as a window to access

all the information supposedly extracted from the BIM model, external databases and estimation results. It also provides ways to interact with that data and perform alterations in the default consideration if needed.

4.3 Relationship and Class Diagrams for the Prototype

The software development process started by charting out different aspects of the tool in a complete Entity-Relationship Diagram (ERD) by using the details provided in the previous section. The first part of this diagram is shown in Figure 4-4, which highlights different entities, their specific attributes and the relationship between these entities. Since, Revit allows the basic modelling of building on the basis of elements, and further detailed modelling is conducted on top of these basic elements. Therefore, the first part of this diagram shows “Elements”, six specialized types of elements namely, “Roofs”, “Beams”, “Foundations”, “Columns”, “Walls”, and “Floors” are typically part of any building project and thus indicated as such in the diagram.

The rudimentary requirement from Revit in our case is material type and quantity attached to the element. This requirement is fulfilled by taking into account the geometry of the element which holds the shape and size of the element and in turn the quantity of the material. Since different instances of same elements such as External and Internal walls may have a different set of materials assigned to it, therefore it is better to assign the material at the geometry level with specific coordinates rather than at the element level. Two representative material types (Concrete and Steel) are studied in this project and are illustrated in the diagram as the specialization of material entity.

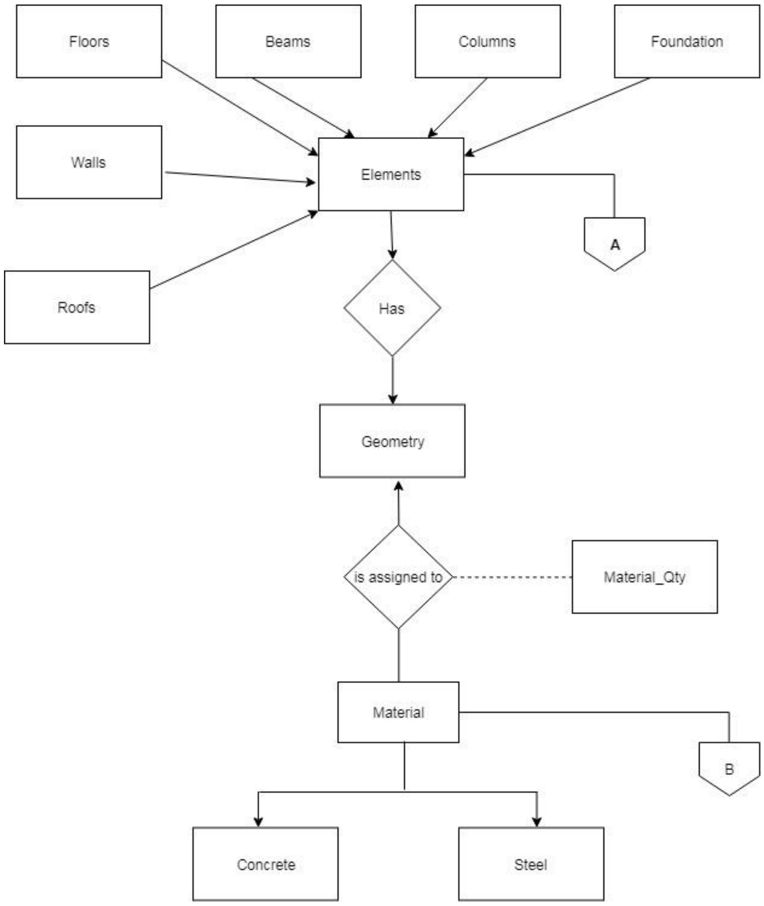


Figure 4-4: ERD for elements, geometry and materials

This ERD continues to Figure 4-5 where the embodied energy is brought into the diagram. A relationship entity "Has" is introduced between the material and "Embodied Energy" as most of the embodied energy content comes from the procurement and transportation of materials. The embodied energy is actually based on three entities "Material Procurement", "Transportation" and "Construction".

To estimate this impact three different databases, need to be accessed apart from the Revit database. Material Procurement entity uses the ICE database, particularly four attributes material name (to match the material assigned in Revit), Sub-type (to further specify the material such as concrete strength again based on the specification in Revit (25Mpa or 40Mpa), density because the embodied energy coefficient is specified as per unit mass basis and Revit provides quantity per volume basis, and finally the embodied energy coefficient as a multiplier to mass for embodied energy content.

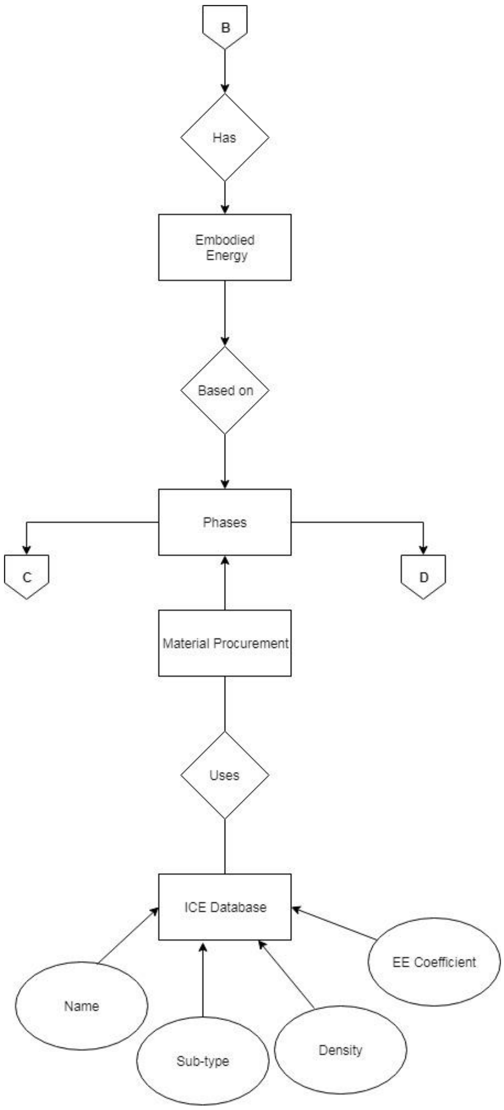


Figure 4-5: ERD for material procurement phase

Figure 4-6 shows the transportation entity which uses the external database holding the transportation tables, particularly Hauling Distance and Vehicle Type. The attributes of interest in the hauling distance are the name of the material as well as distance of the material supplier from the project site. The vehicle is chosen based on its capacity as well as type of material it can haul. These attributes are summed up as the attribute named as Capacity. The Fuel Consumption is pivoted on the type and capacity of the vehicle and represents mileage of that vehicle. The fuel coefficient is actually the lower heating value of the fuel or simply the energy per litre of fuel.

The construction entity uses the Equipment List based on the element to be constructed. The equipment list further has an Equipment work rating entity which holds attributes like equipment rated-power and the work load per day. The equipment rated power is work output per unit energy. Work load per day is total work done by the equipment per day based on the up-time of the equipment and hours of work in a day.

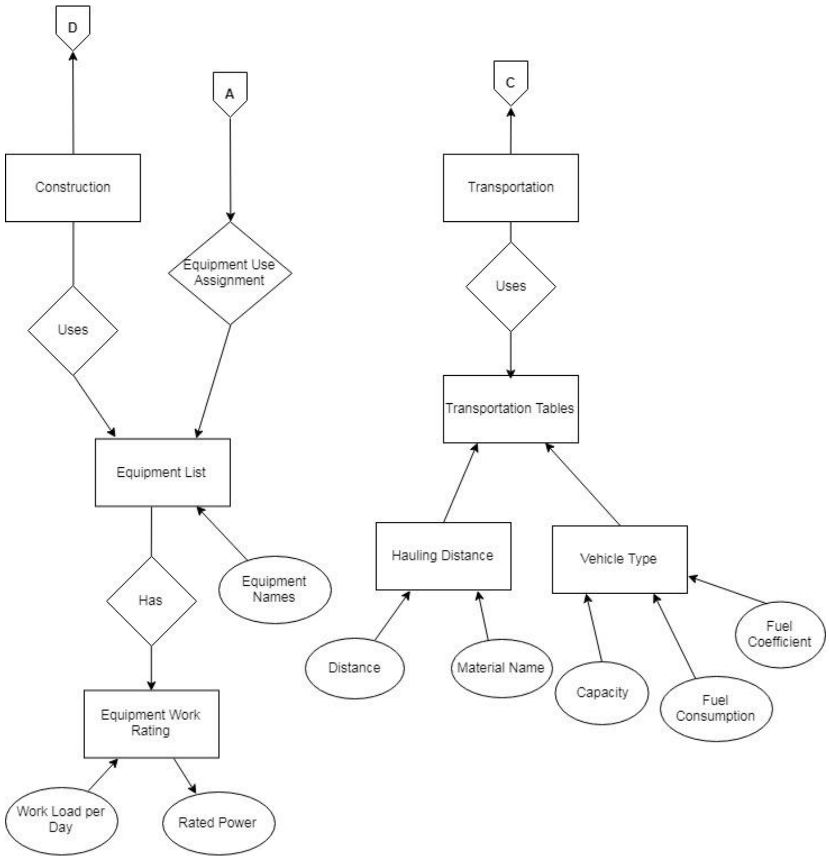


Figure 4-6: ERD for transportation and construction phase

A Unified Modelling Language (UML) class diagram is formed on the basis of the ERD to visualize the design of the tool is shown in Figure 4-7. This diagram was formed based on the bottom-up approach, keeping in view the end goal of estimating the total embodied energy and then expanding the system to include all the aspects. So, a class of Embodied_Energy was created with three public variables holding the energy values obtained by different phases of the project. Apart from that, three methods were also included to calculate the total energy, generate result tables and consequently graphs representing the results. This class then seeks three other classes, namely MaterialEmbEnergy, TransportationEnergy and ConstructionEnergy to feed in the values for material, transportation and construction energy respectively.

MaterialEmbEnergy is an abstract class with six implementations or sub-classes of Mat_Emb_Walls, Mat_Emb_Columns, Mat_Emb_Floors, Mat_Emb_Foundations, Mat_Ebm_Roofs and Mat_Emb_Beams. This is necessary because Revit holds all the data on the basis of elements and not materials. Subsequently, to get to the exact quantity of material used in different layers of an element, the elements must be collected based on its type (such as wall or column), and only then their properties can be accessed . Therefore, the methods collect_elements() and identify_material() needs to be formulated within this class. The method fill_subtype() handles further specification of material which later helps in shortlisting the material from ICE database.

Once the material is shortlisted, get_ICE_parameters() method searches for the specific material type in ICE and returns values such as ee_coefficient. The cal_mass_energy() method just converts volume to mass using density. Finally, cal_emb() method determines the total embodied energy of the material used in that element. One external class SqlDB and one external attribute m_doc are also part of this class. SqlDB helps to query data from the external database via

Sql_Connection() and m_doc aids in accessing the Revit modelling database. A separate Command class holds the methods to access the Revit model and distributes this access through m_doc.

TransportationEnergy first imports all the quantities calculated in the previous class by accessing the public attribute material_qty. Since the transportation interface needs to have dynamic tabs based on all the unique materials used on the project, a crt_dyn_mat_tab() method executes the same. All the material used in different elements needs to be summed up in one quantity that will determine the transportation impact of that material, so a cal_mat_qty() method is created. Now, this material needs to be supplied from the supplier site which will have an associated distance, so the fill_mat_distance() method is written to match the material name in the external database and return the distance that will be stored in mat_haul_dist. The methods fill_veh_typ() and get_veh_para() help in shortlisting the vehicle most suitable to transport a particular material depending on its form and quantity. The concept of round trip is important in the domain of transportation and this notion of going back to bring something constitutes almost fifty percent of the impact. This is taken care of by the method cal_rnd_trp().

ConstructionEnergy is also an abstract class with six implementations or sub-classes of Cons_Eng_Walls, Cons_Eng_Columns, Cons_Eng_Floors, Cons_Eng_Foundations, Cons_Eng_Roofs and Cons_Eng_Beams. The ConstructionEnergy class also imports the quantity of material from the MaterialEmbEnergy class. The get_equip_list() method gets the list of equipment required to construct the respective element from the external database. The get_equip_info() method looks for the rated power and workload per day information for the external database. The cal_work_day() method determines the total hrs of work conducted by the equipment based on the elements to be installed. The methods of chng_equip_config() and chng_work_hr() enables to deviate from the default considerations of equipment list per element and work hours per day respectively.

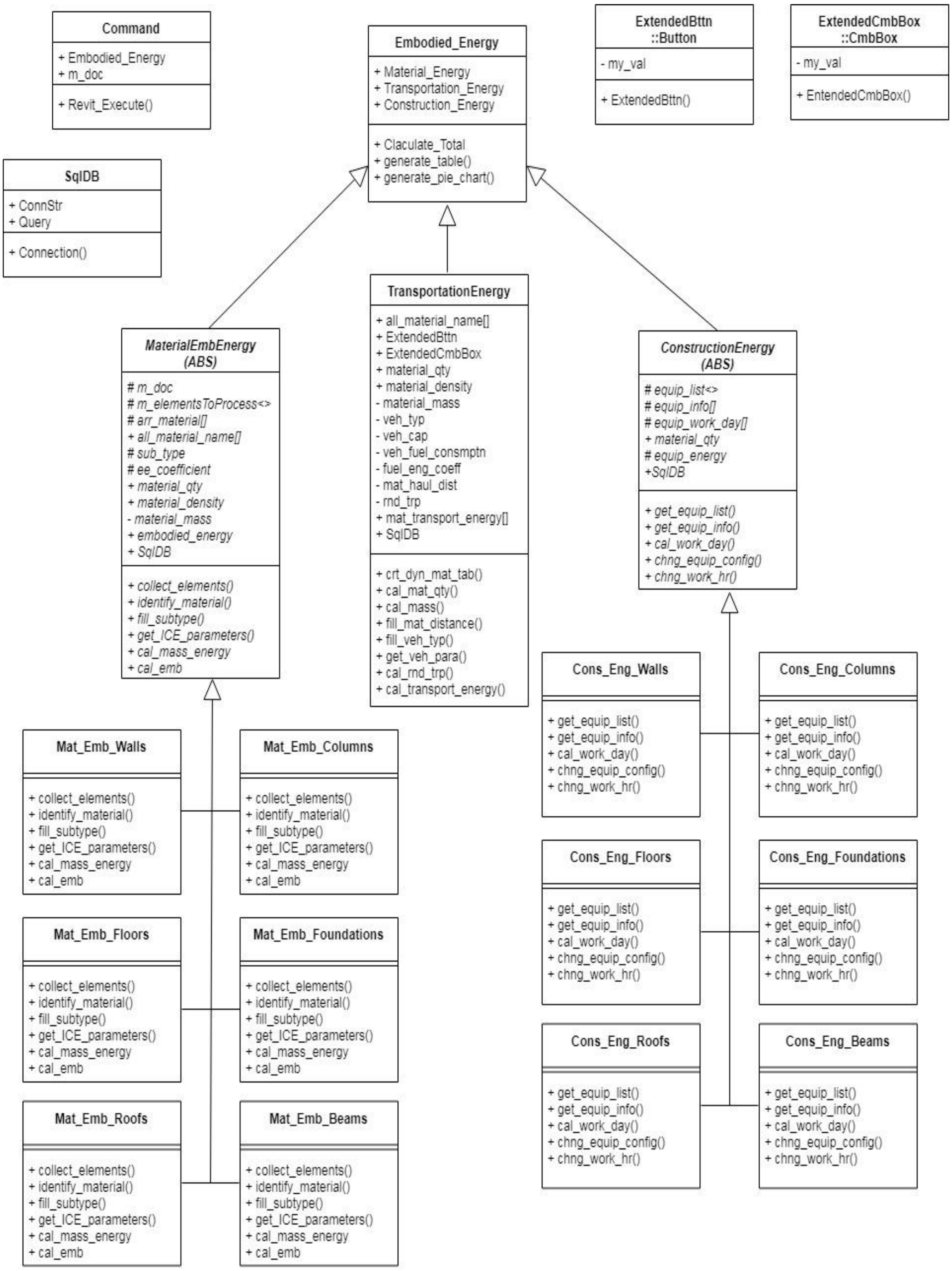


Figure 4-7: UML Class Diagram

4.4 Extension of IFC to Incorporate Embodied Energy Impact

As discussed in section 3.5.3 of the methodology, the cost embedment was found helpful for the integration of embodied energy in the IFC4 model. Within the fundamental concepts of the IFC4 model, a concept of Resource Cost is defined as "Resources can have associated costs indicating financial costs and environmental impacts incurred according to a specified base quantity" (IFC4, e). However, this template is fully implemented only for the financial cost aspect throughout the breadth and width of the IFC model. Nevertheless, this template paves way for integration of embodied energy concepts in the IFC model.

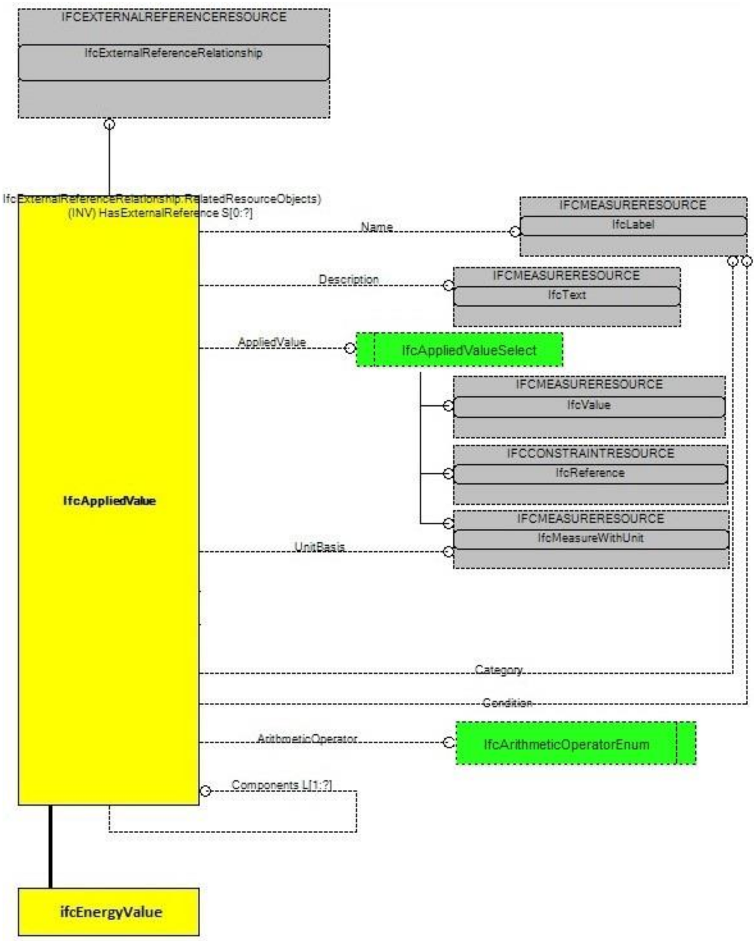


Figure 4-8: Express-G diagram of the newly proposed *IfcEnergyResource*

Looking at the lowest layer of the IFC Model in Table 2-3, an entity of *IfcCostValue* represents the basic unit of cost. A similar entity for energy with the name "*IfcEnergyValue*" is proposed to describe the fundamental energy unit. *IfcArithmaticOperatorEnum* remains same as with the cost concept, but the *IfcAppliedValue* is modified by removing the date time association as "time value of money" concept is not relevant in the context of energy. Also, the concept of *IfcCurrencyRelationship* is irrelevant in the current context. Figure 4-8 shows the Express-G diagram for this newly proposed entity of "*IfcEnergyValue*" under the new resource of "*IfcEnergyResource*".

The next layer is the Core Layer, with *IfcControl* entity. Since, energy controls or constraints the utilization of products, processes or resources; thus, fits well in the definition of *IfcControl*. Within the interoperability layer two new energy related entities are proposed: *IfcEnergyItem* and *IfcEnergySchedule.* Two new data types: *IfcEnergyItemTypeEnum* and *IfcEnergyScheduleTypeEnum* are also proposed. The Express-G diagram for these newly proposed entities and types are shown in Figure 4-9.

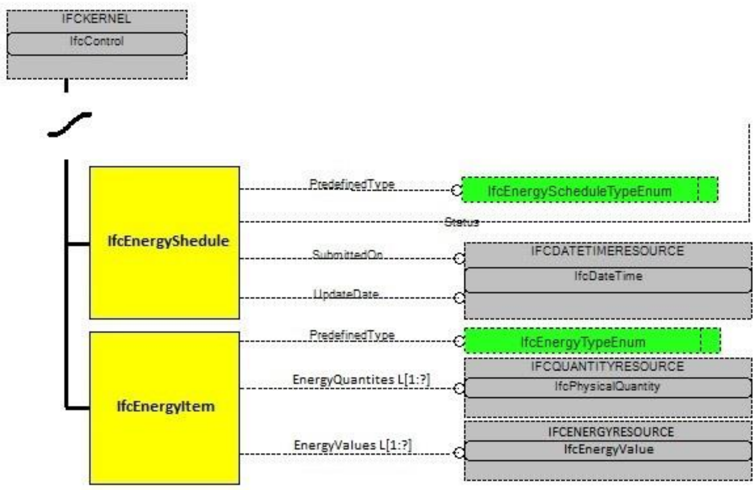


Figure 4-9: Express-G diagram for *IfcEnergySchedue* and *IfcEnergyItem*

The domain layer is the top most layer of the IFC schema. An extension in this domain is proposed to include *IfcEnergyResource* as an optional attribute to account for the embodied energy impact of the construction resources. The Express-G diagram for the modified *IfcConstructionMgmtDomain* is shown in Figure 4-10.

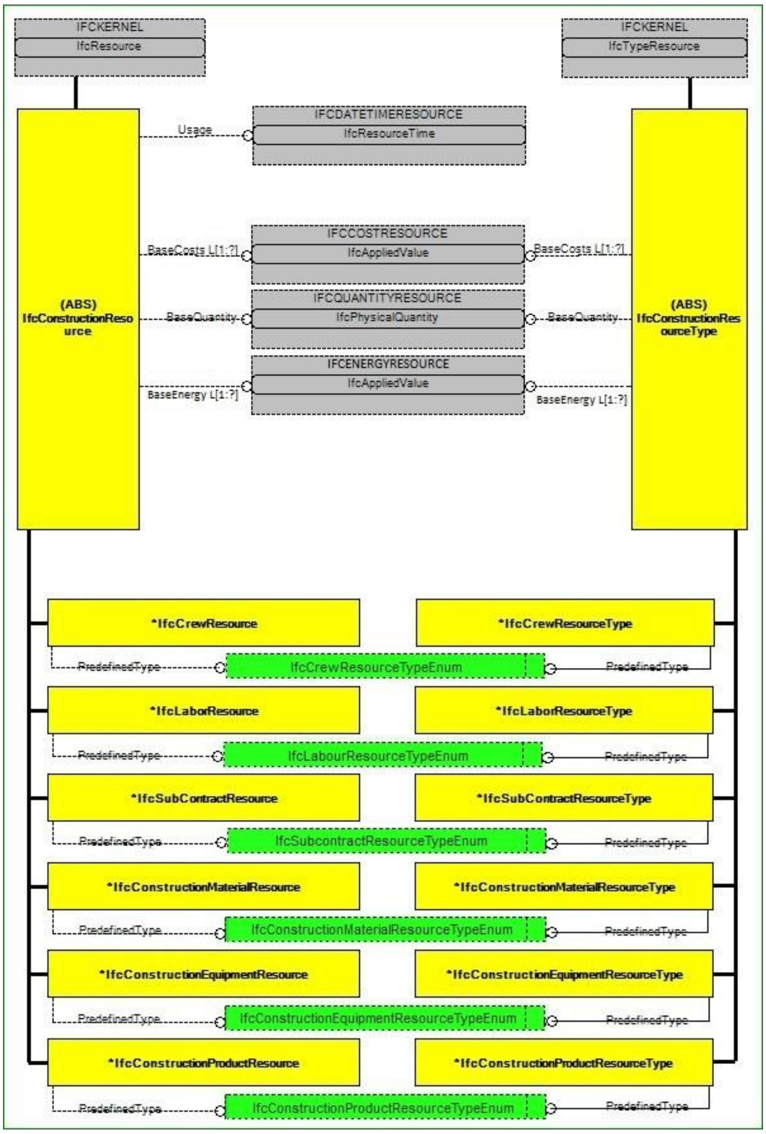


Figure 4-10: Express-G Diagram for the modified *IfcConstructionMgmtDomain*

In this manner, the theoretical extension of the IFC4 model for incorporating the embodied energy impact is complete. Table 4-1 shows the summary of the proposed entities for energy embedment in the IFC model. The implementation of this methodology is presented in section 5.4 by using an open source BIM software development tool.

Table 4-1: Proposed entities for extension of IFC4 model to incorporate embodied energy

IFC Layer	IFC Layer component	IFC Entity and Types
Domain Layer	*IfcConstructionMgmtDomain*	*IfcConstructionResource* *IfcConstructionResourceType*
Interoperability Layer	*IfcSharedMgmtElements*	**IfcEnergySchedule* **IfcEnergyItem* **IfcEnergyItemTypeEnum* **IfcEnergyScheduleTypeEnum*
Core Layer	*IfcKernel*	*IfcControl*
Resource Layer	**IfcEnergyResource*	**IfcEnergyValue* *IfcArithmeticOperatorEnum* *IfcAppliedValueSelect*

* represents new entities

4.5 Summary

This chapter describes the development process for the prototype. The development process starts with declaration of three different workflows for three kinds of estimations i.e. Material, Transportation and Construction energy. To realize these workflows database diagrams are developed. Initially the entity relationship diagrams are developed around six basic elements (Foundation, Column, Beam, Floor, Roof and Wall) and their interaction with the external databases (ICE, Transportation and Construction Equipment). Secondly, the UML diagram is developed to represent entities in the ERD as parameters in software as well as the functions that can be performed over these parameters. Finally, a method is presented to export the embodied energy values in the IFC format to retain the results in the BIM environment.

CHAPTER 5: IMPLEMENTATION

5.1 Embodied Energy Assessment

Once the model is loaded in the Revit software the user interface of the tool can be launched through the add-ins menu. The interface has four tabs: Material Energy, Transportation Energy, Construction Energy and Embodied Energy. Figure 5-1 shows the contents of the Material Energy tab, consisting of six sub-tabs representing different common building elements such as walls, columns, beams, etc. Within each tab there is a combo box which is automatically populated with all the materials used in the current project for that building element. For each material in the list there is a subtype; for example, concrete's subtype is based on its strength as it will dictate the embodied energy coefficient appropriate for it.

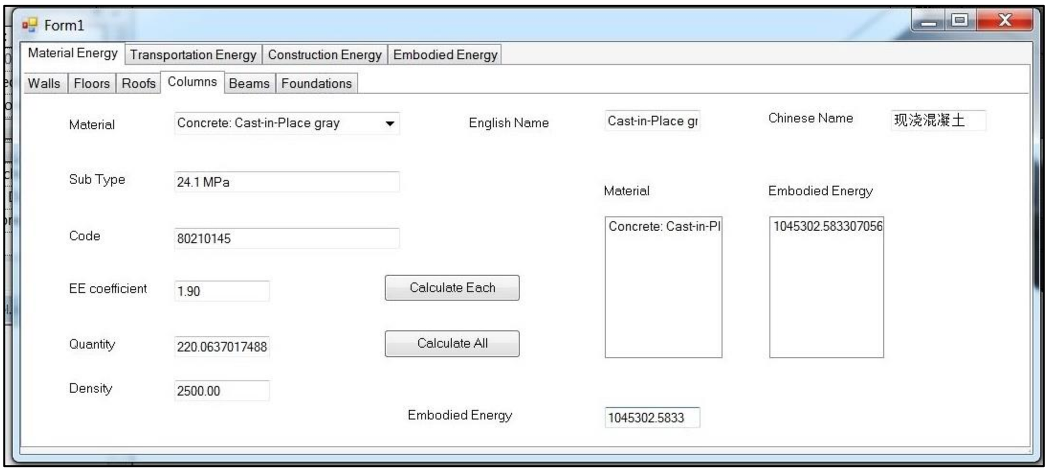


Figure 5-1: Material Embodied Energy Tab

Once the appropriate embodied energy coefficient is selected from the ICE database, the density and quantity of that material is extracted from the model and finally the embodied energy for that material used in element is calculated. The user has the liberty to either calculate the material's embodied energy one by one and may edit any value as deemed necessary or calculate all the material embodied energy together by considering the values suggested by the software.

Figure 5-2 shows the contents of the transportation energy tab; the tool automatically generates separate sub-tabs for each of the unique materials used in the whole project. The user can choose from the available transportation vehicles and their capacities, and by using the material quantity and capacity of the vehicle, the tool calculates the roundtrips required to transport the selected material. From the suppliers' database, the hauling distance from the factory to the site for that material is queried. The other required parameters such as vehicle fuel consumption per unit distance and fuel energy coefficient are selected from the designated databases based on the choice of vehicle.

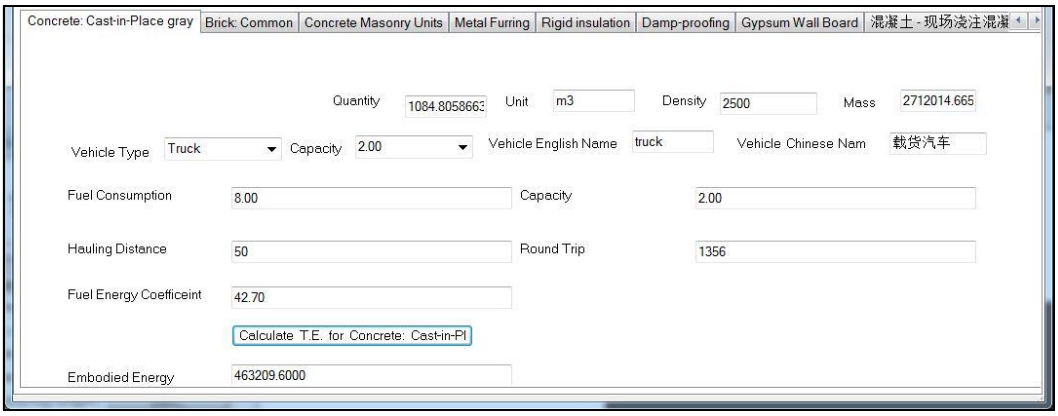


Figure 5-2: Transportation Energy Tab

Figure 5-3 illustrates the contents of the Construction Energy tab. Like the Material Energy tab, it consists of 6 sub-tabs representing standard work packages. For each work package, a unique set of equipment is already set by default. The equipment entities are made intelligent in terms of their function to select the associated quantity from the material quantity information. For instance, in the context of foundations, the Excavator looks for the depth and type of foundation to be installed for calculating its work load. Consequently, in the same context, the Concrete Mixer looks for the quantity of concrete to be processed. To make the tool user friendly and work with minimum inputs, all this information is not displayed in the main interface. Nevertheless, to make the tool flexible all this information, along with the power ratings of equipment, are made available to access and modify under the “Change Equipment Configuration” button. The user may also add/delete any equipment for the respective work package using the right click button. Daily productivity, being readily available information in the context of construction equipment, is made available to access and modify on the main interface. The user interface of the Embodied Energy tab is discussed under section 5.1.

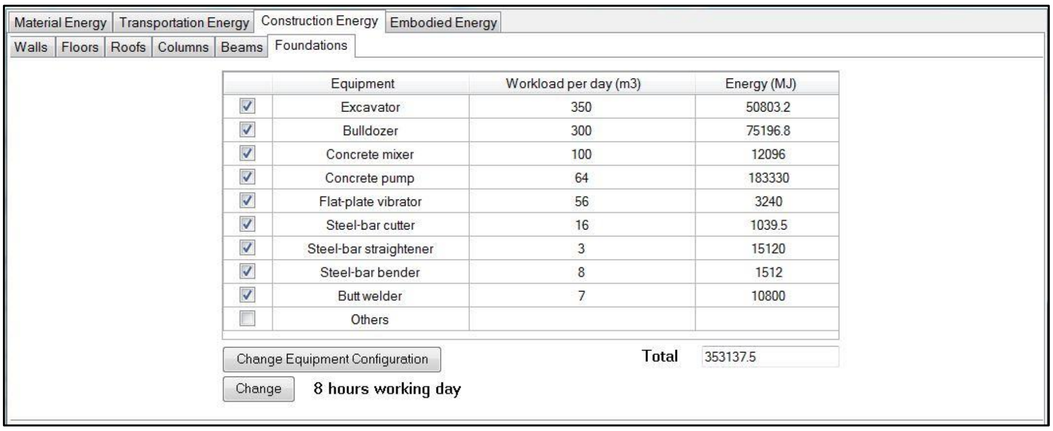


Figure 5-3: Construction Energy Tab

5.2 Simulation of Transportation Energy

As proposed in the methodology, simulation models for transportation and construction activities are developed for optimization purposes. Different processes were modelled by using the simulation software package Simphony to analyze the potential of reducing energy consumption for transportation and construction. Two examples are shown here for transportation of steel rebars and placement of ready mixed concrete. Figure 5-4 shows the simulation model for transportation of steel rebars. It primarily involves four activities (Load, Travel, Unload and Travel Back) and two resources (Loader and unloader).

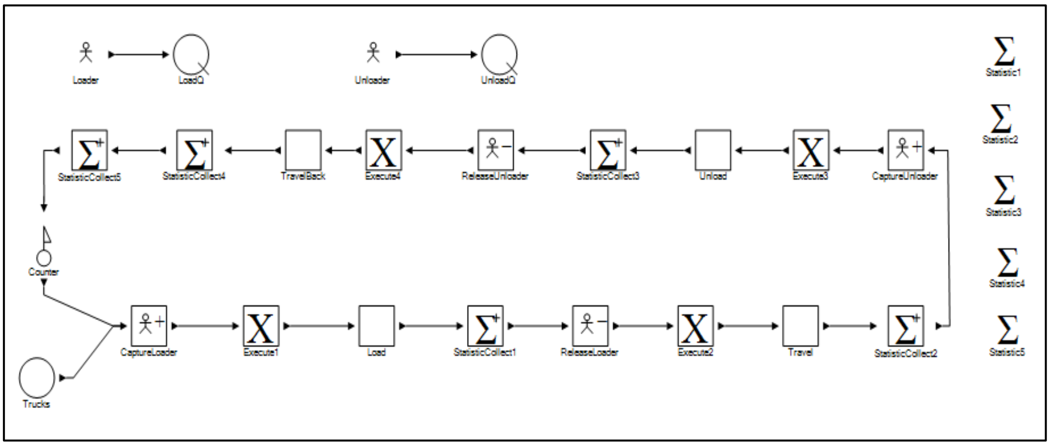
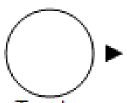
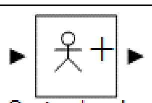
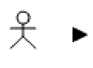
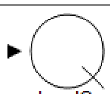
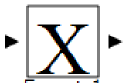
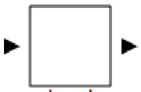
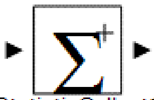
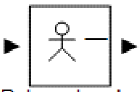


Figure 5-4: Simulation model for transportation of steel rebars

Before providing details for this specific model, there is a need to define the individual elements seen in the model with their significance. Following table 5-1 lists all the elements used in the model with their names and descriptions.

Table 5-1: Details of the elements used in simulation model [59].

Element Shape	Name and description
 Trucks	Create Element: Responsible for creating entities and introducing them in the model. It can control the number of entities running in the model (in our case trucks).
 CaptureLoader	Capture Element: Responsible for granting the exclusive use of one or more servers of a Resource to an entity. This element checks if a server is available, if not, it ensures that entity is moved to a waiting file (queue) for a resource.
 Loader	Resource Element: Responsible for defining a shared resource. It enables to assign the number of servers as well as checking the utilization of the resource.
 LoadQ	File Element: Responsible for defining a queue in which entities wait for a shared resource.

 Execute1	Execute Element: Runs a snippet of user written code whenever an entity arrives at its input point. In this study it is used to note the time an entity enters this element.
 Load	Task Element: Responsible for modelling an activity such as load, unload, travel, pour etc. A duration for the activity can be set through this element.
 Statistic1	Statistic Element: Responsible for defining a custom statistic such as calculation of cost or cycle time. In this study this element is used to calculate both time and energy.
 StatisticCollect1	StatisticCollect Element: Responsible for adding a single observation to a Statistic element. Since, this element is programable, so it allows how that static is added once an entity passes through it and thus enables the calculation of energy with respect to time.
 ReleaseLoader	Release Element: Allows an entity to return servers it has previously captured to the pool of available servers.
 Counter	Counter Element: Used to record important milestones in the lifecycle of an entity. It enables to terminate the model by providing upper limit to the production. It also aids in setting the contribution of each entity towards the production. Finally, it helps in calculating the production rate.

Being a transportation focused model, the basic model entities are trucks. Two kinds of variations are performed in this simulation model: Variation in the number of trucks and size of trucks. The variation in the number of trucks is set by changing the quantity attribute of the create element as shown in the Figure 5-5.

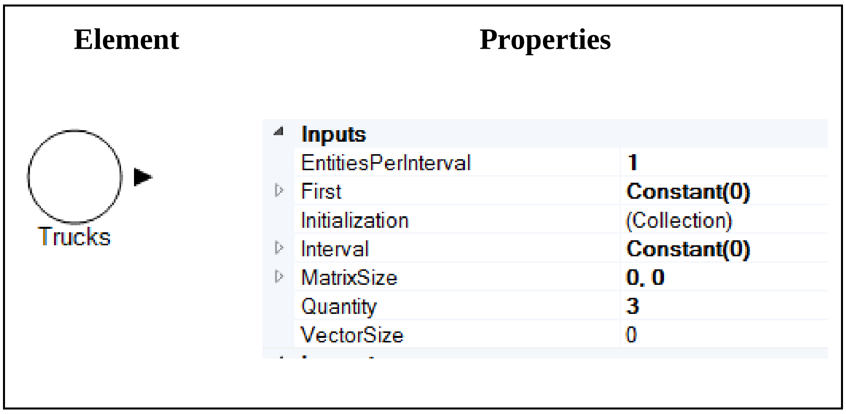


Figure 5-5: Properties of the Create element

Once the quantity is set, the process starts with an event that a truck captures a loader resource. This loader resource loads up the truck with the steel bars and the time is noted by "StatisticCollect1". Three things are important to note here; recording the time when the entity starts to load, setting the duration of the load activity and the calculation of energy consumed to perform the activity. To record the time when the activity started to perform, an execute element is introduced with the code as shown in the Figure 5-6.

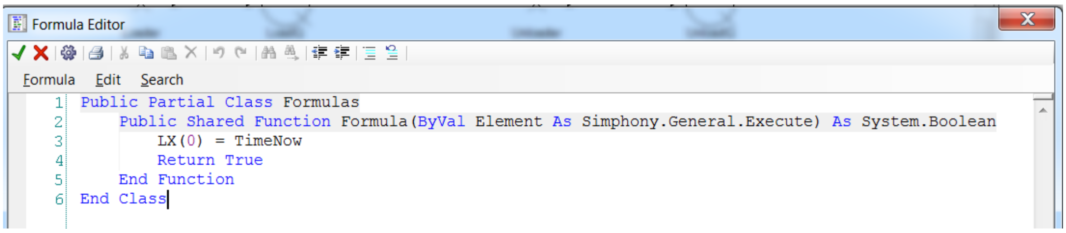


Figure 5-6: Code snippet for the Execute element

Here the variable LX(0) stores the time at every instance an entity passes through it. For setting the duration of the load activity, two types of inputs are required: the probability distribution type of the duration such as triangular, exponential etc. and the numerical input as required by the distribution. Figure 5-7 shows the input for the load activity duration.

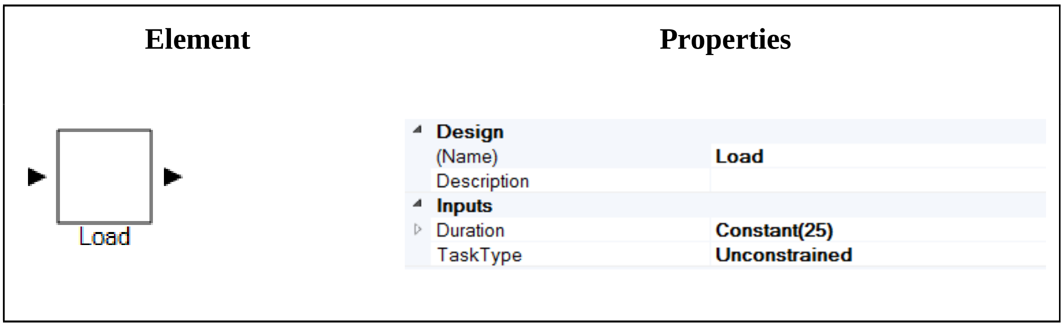


Figure 5-7: Properties of the Load element

Here the duration is set as 25 minutes and the distribution is set as constant. This duration depends upon the truck size and discussion for this duration is presented in the results section. As far as the calculation of energy is concerned, this is performed within the StatisticCollect element. As mentioned previously the StatisticCollect element is used to record the cycle time, if multiplied by the coefficient representing energy consumption per unit time, it can provide the energy consumed to perform an activity by a single entity. Figure 5-8 shows the code snippet for the StatisticCollect.

Element	Code Snippet
	LX(1)= TimeNow-LX(0) GX(0) = GX(0) + LX(1) GX(1) = GX(0)* 4.3554 Return GX(1)

Figure 5-8: Code snippet for the StatisticCollect element

There are three variables LX(1), GX(0) and GX(1). LX(1) calculates the interval between the start and finish of activity which in turn calculates the up time of the equipment used to conduct the activity, in this case the loader. GX(0) calculates the cumulative up time for the equipment and GX(1) calculates the energy consumed by the equipment. To obtain the coefficient which converts time to energy needs some conversation equation. The following equation 5.1 can aid in the understanding this conversion.

$$C1 = \frac{0.215\text{Kg}}{\text{KWH}} \text{X} \frac{1\text{ L}}{0.84\text{ KG}} \text{X } 24\text{ KW X } \frac{1\text{H}}{60\text{Min}} X \frac{42.7MJ}{L} = 4.3554 \frac{MJ}{Min} \quad (5.1)$$

Where:

0.215Kg/KWH is the fuel consumption for the loader running at loaded state
0.84 KG/L is density of diesel fuel
24 KW is the rated power for loader
42.7 MJ/L is the lower heating value of fuel

Since there are limited number of resources, a resource when not in use should be released using the Release Element. Now, this makes a complete set of elements required to calculate the energy consumption for a particular activity. By looking at the model presented in the Figure 5-4, it can be observed that a set pattern of elements is repeated for all the activities; it starts with an Execute Element which notes down the time and if the resource is need then a Capture Element is used, followed by activity element (such as Load, Travel, Unload and Travel Back) and finally StatisticCollect and Release Element if resource was captured previously.

So, after completing the load activity, the truck then travels to reach the construction site. The "StatisticCollect2" notes the time it took to reach the destination. And to convert the duration to associated energy consumption, a coefficient C2 is calculated as shown in the equation 5.2.

$$\text{C2} = \frac{0.15\text{ L}}{\text{KM}} \text{X} \frac{30\text{KM}}{\text{H}} \text{ X } \frac{1\text{H}}{60\text{Min}} \text{X} \frac{42.7\text{MJ}}{\text{L}} = 3.2025 \frac{\text{MJ}}{\text{Min}} \quad (5.2)$$

Where:

0.15L/KM is the fuel efficiency for 8 ton truck, 0.17 for 10 ton and 0.19 for 12 ton
30 KM/H is the average speed of the truck when loaded
42.7 MJ/L is the lower heating value of fuel

To be unloaded, the truck needs an unloader to remove all the steel bars and therefore the truck captures an unloader. This act of capturing unloader depends on the availably of the unloader resource and thus may cause waiting time. The "StatisticCollect3" notes down the simulated time for this activity. To convert the time to energy a same coefficient C1 which is calculated previously in equation is used as equipment is same. Once unloaded the truck travels back to pick up more steel bars, it has one change that the speed has increased as now the truck would be in unloaded state. So, equation 5.3 illustrates the calculation for the coefficient C3 to convert time to energy for "StatisticCollect4".

$$C3 = \frac{0.15\ \mathrm{L}}{\mathrm{KM}} \mathrm{X} \ \frac{60\mathrm{KM}}{\mathrm{H}} \ \mathrm{X} \ \frac{1\mathrm{H}}{60\mathrm{Min}} \mathrm{X} \frac{42.7\mathrm{MJ}}{\mathrm{L}} = 6.405 \frac{\mathrm{MJ}}{\mathrm{Min}} \tag{5.3}$$

Where:

0.15L/KM is the fuel efficiency for 8 ton truck, 0.17 for 10 ton and 0.19 for 12 ton
60 KM/H is the average speed of the truck when unloaded
42.7 MJ/L is the lower heating value of fuel

"StatisticCollect5" estimates the sum of the energy consumption for different activities. Before reaching the factory, a counter counts the trips and decides if any additional trips are required. This counter actually determines the termination point of the simulation model. The figure shows the inputs required to set up the counter. Two values are of importance here, Limit and Step. The limit sets the terminal count, which actually depends on the total material to be hauled and Step is hauling capacity of each hauling unit. For instance, the Figure 5-9 shows the input for 563 tons of material hauling by a 12-ton truck.

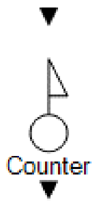
Element	Properties	
	**Inputs**	
	Initial	0
	Limit	**563**
	ReportStatistics	**True**
	Step	**12**

Figure 5-9: Properties for the counter element

To understand how the results are interpreted and represented for this study, a sample is presented in this section. The initial output from the software is presented in the Figure 5-10 below. These results are for three 8-ton trucks introduced in the model.

Non-Intrinsic Statistics

Element Name	Mean Value	Standard Deviation	Observation Count	Minimum Value	Maximum Value
Scenario1 (Termination Time)	3,265.000	0.000	1.000	3,265.000	3,265.000
Statistic1	4,028.745	2,294.349	73.000	108.885	7,948.605
Statistic2	7,013.475	3,993.378	72.000	192.150	13,834.800
Statistic3	3,179.442	1,810.331	72.000	87.108	6,271.776
Statistic4	6,917.400	3,937.903	71.000	192.150	13,642.650
Statistic5	21,362.064	11,892.442	71.000	1,269.471	41,697.831

Counters

Element Name	Final Count	Production Rate	Average Interarrival	First Arrival	Last Arrival
Counter	568.000	0.174	44.714	135.000	3,265.000

Resources

Element Name	Average Utilization	Standard Deviation	Maximum Utilization	Current Utilization	Current Capacity
Loader	55.9 %	49.7 %	100.0 %	0.0 %	1.000
Unloader	44.1 %	49.7 %	100.0 %	0.0 %	1.000

Waiting Files

Element Name	Average Length	Standard Deviation	Maximum Length	Current Length	Average Wait Time
LoadQ	0.023	0.194	2.000	0.000	1.027
UnloadQ	0.000	0.000	1.000	0.000	0.000

Figure 5-10: Results for the transportation model

The mean value of the Scenario1 (3265 minutes) represents the total time taken in minutes to finish the complete hauling. The maximum value of Statistic5 (41,697.831 MJ) represents the total energy consumption. The production rate is given under the counter (0.174 Tonnes / min). 55.9 and 44.1 percent of utilization is found for the loader and unloader resource respectively. By clicking on the individual elements further investigation can be performed with respect to time. The following Figure 5-11 shows the graph of Production Rate versus Simulation Time.

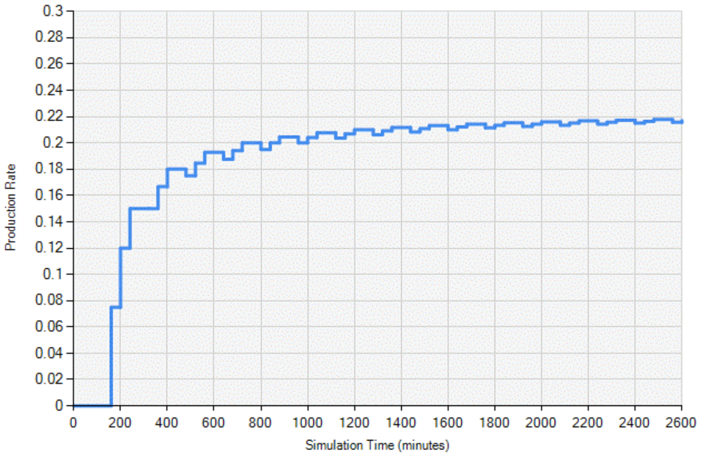


Figure 5-11: Production rate of the transportation model

Similarly, the utilization of the resources (loader and unloader) with respect to time can be seen in Figure 5-12 and Figure 5-13 respectively.

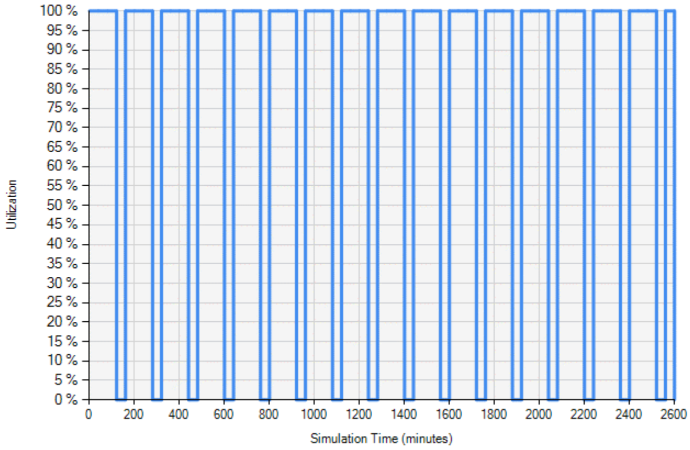


Figure 5-12: Utilization of Loader Resources

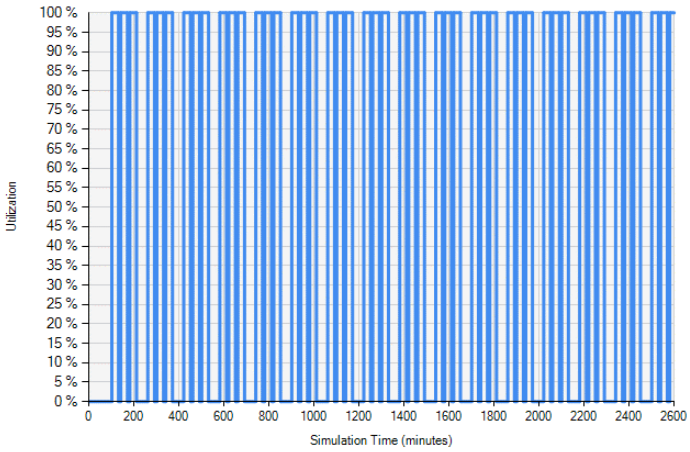


Figure 5-13: Utilization of Unloader Resources

5.3 Simulation of Construction Energy

Another model for placement of ready mix concrete is shown in Figure 5-14. It involves both the transportation of ready mix concrete trucks to the site as well as pouring of the concrete on the site using pump. This model has four activates as well (Loading ready-mix concrete trucks, Travel, Pouring concrete on site using Pump and Travel Back) and two resources (Batching Plant of concrete and Concrete pouring pump).

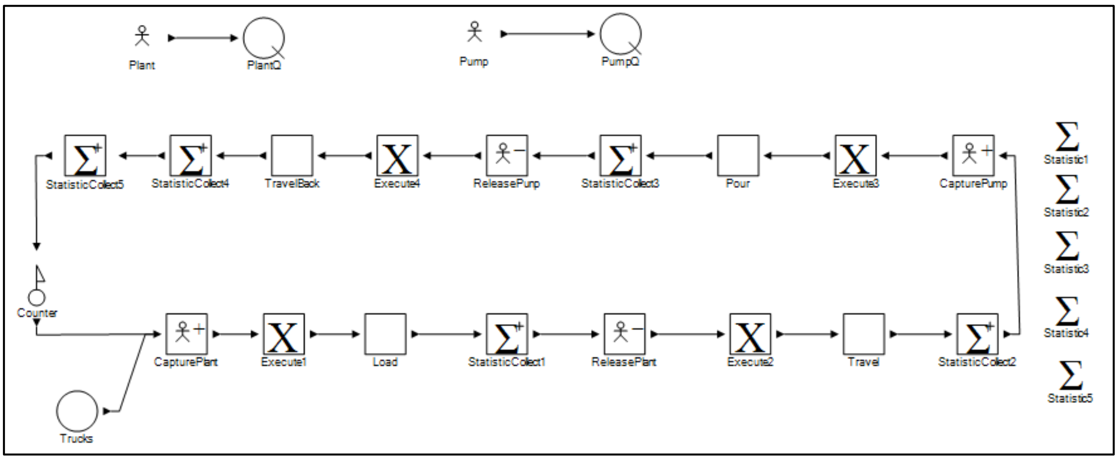


Figure 5-14: Simulation model for placement of ready mix concrete

The basic entities in this model are also trucks as actual production is governed by the number of trucks of concrete poured on site. The process starts with an event that a truck captures a plant resource, this plant resource fills up the truck with the concrete and the time at this point is noted by StatisticCollect1. Once the plant resource fills up the truck it is released to serve other trucks, and the truck then travels to the site. StatisticCollect2 notes the time taken to reach the construction site. To convert time to energy a coefficient C4 is calculated in equation 5.4.

$$C4 = \frac{0.5\ \mathrm{L}}{\mathrm{KM}} \mathrm{X}\ \frac{30\mathrm{KM}}{\mathrm{H}}\ \mathrm{X}\ \frac{1\mathrm{H}}{60\mathrm{Min}} \mathrm{X} \frac{42.7\mathrm{MJ}}{\mathrm{L}} = 10.675 \frac{\mathrm{MJ}}{\mathrm{Min}} \qquad (5.4)$$

Where:

0.5 L/KM is the fuel efficiency
30 KM/H is the average speed of the truck when loaded
42.7 MJ/L is the lower heating value of fuel

Then the truck looks for a pump resource, once found idle, the pump resource is captured by the truck and concrete starts to pour. StatisticCollect3 notes the time taken to pour the concrete. Equation 5.5 calculates the coefficient C5 to convert time to energy.

$$C5 = 97\mathrm{KW}\ \mathrm{X}\ \frac{3.6\mathrm{MJ}}{\mathrm{KWH}}\ \mathrm{X}\ \frac{1\mathrm{H}}{60\mathrm{Min}} = 5.8212 \frac{\mathrm{MJ}}{\mathrm{Min}} \qquad (5.5)$$

Where:

97 KW is the rated power of pump
3.6 MJ/KWH is the conversion unit for KWH to MJ

Once the concrete is poured, the pump resource is released to pour concrete from other trucks. The truck then sets to return back to the ready mix-concrete plant to get filled up. The StatisticCollect4 notes down the time taken to travel back and equation 5.6 calculates the coefficient C6 to convert time to energy. A counter is placed to count the number of trucks and decide if any additional trucks are required or the simulation should terminate.

$$C6 = \frac{0.5\ \mathrm{L}}{\mathrm{KM}} \mathrm{X}\ \frac{60\mathrm{KM}}{\mathrm{H}}\ \mathrm{X}\ \frac{1\mathrm{H}}{60\mathrm{Min}} \mathrm{X} \frac{42.7\mathrm{MJ}}{\mathrm{L}} = 21.35 \frac{\mathrm{MJ}}{\mathrm{Min}} \qquad (5.6)$$

Where:

0.5 L/KM is the fuel efficiency
60 KM/H is the average speed of the truck when unloaded
42.7 MJ/L is the lower heating value of fuel

An application of these models on a case study is presented in section 6.3.

5.4 IFC Extension

5.4.1 XBIM Toolkit

To implement the methodology for extension of IFC to incorporate embodied energy content in the IFC model a .NET open source software development toolkit for BIM, eXtensible Building Information Modelling (xBIM) Toolkit [80] is used. It enables the software developers to create, read and view IFC based BIM models using any .NET language and environment. These models, once created or updated, remain operable with other BIM authoring tools. The Figure 5-15 below shows all the libraries and namespaces within the libraries for the Xbim Toolkit.

The xBIM toolkit has six libraries namely, XbimEssentials, XbimWindowsUI, XbimGeometry, XbimExchange, XbimCobieExpress and XbimWebUI. However, XbimEssential acts as the foundational library holding all the foundational components of Xbim.

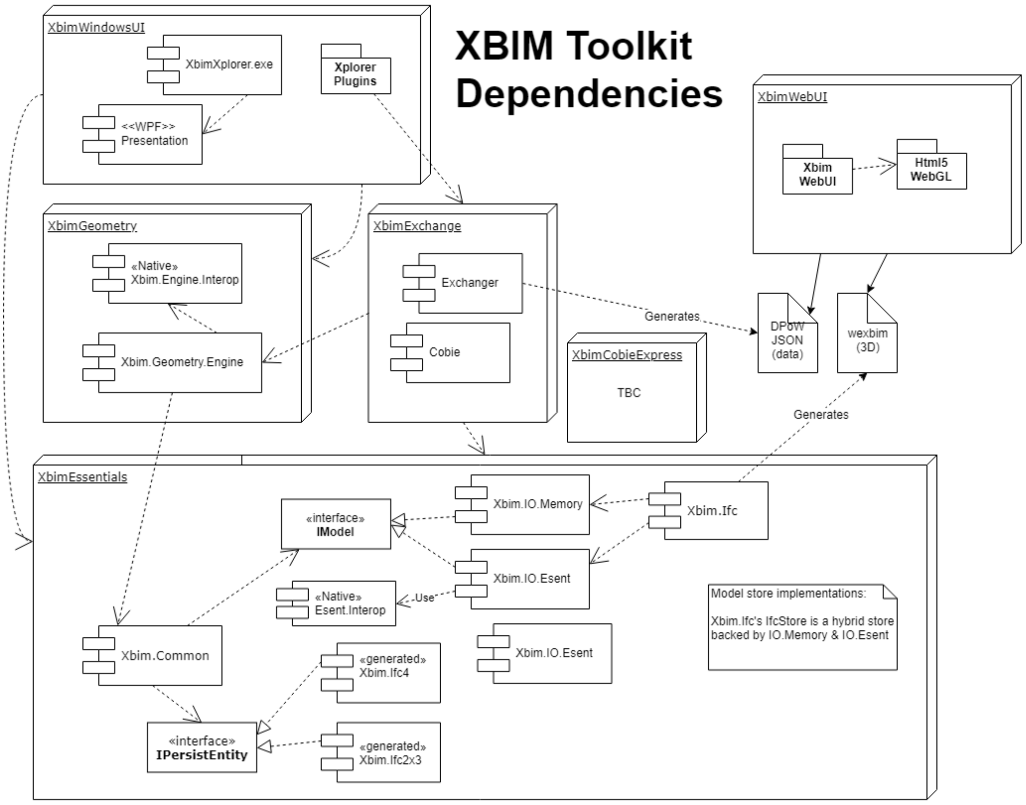


Figure 5-15: Architecture of XBIM Toolkit [80]

Closely observing the XbimEssential core library can lead to the understanding that Xbim.Common namespace plays a fundamental role in generating the IFC models. This namespace is connected to two interfaces, IModel and IPersistEntity. The implementation of this IPersistEntity generates two versions of IFC models IFC2X3 and IFC4. This study is focused on the IFC4 implementation and thus restrict the discussion to the IFC4 version. The Figure 5-16 below shows some of the namespaces within the IFC4 namespace, these namespaces represent the same entities shown in Figure 3-5. And the Figure 5-16 also shows the classes inside the cost resource.

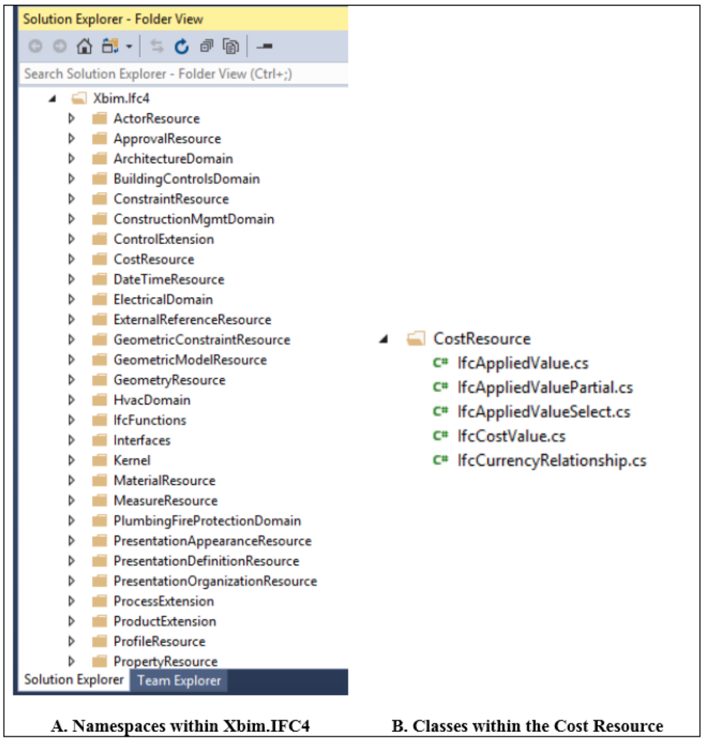


Figure 5-16: Namespace and Classes within XBIM.IFC4

5.4.2 New Classes developed for Embodied Energy Representation

In order to implement the methodology described in section 4.4, the proposed entities mentioned in Table 4-1 are created in the Xbim.IFC4 namespace of the xBIM Essential Library. First a class of *IfcEnergyValue.cs* is created within the namespace of *Xbim.Ifc4.EnergyResource.* The function of this class is to provide a concrete implementation of the abstract class *IfcAppliedValue.cs* which exists within the namespace of *Xbim.Ifc4.CostResource.* In practice, this caters to the need of defining an independent energy unit that can later be summed to find the total energy. Secondly, two *enum* data types are defined by using the classes *IfcEnergyItemTypeenum.cs* and *IfcEnergyScheduleTypeEnum.cs* within the namespace of *Xbim.Ifc4.Interfaces.* Finally, two more classes (*IfcEnergyItem.cs* and *IfcEnergySchedule.cs)* are developed within the namespace of *Xbim.Ifc4.SharedMgmtElements.*

Again, these classes act as concrete implementations of the abstract class of *IfcControl.cs* which exist in the namespace *Xbim.Ifc4.Kernel*. These classes enable storage of embodied energy data for individual resource such as material or any construction resource. *IfcEnergyItem* represent energy for an individual instance and *IfcEnergySchedule* consolidates different *EnergyItems* in a contextual dimension such as base plan, actual or simulated. Figure 5-17 shows all the new classes developed and Figure 5-18 shows the interaction of the new entities with the previously existing higher order entities. A detail of the use case for these newly developed entities is presented in the case next section.

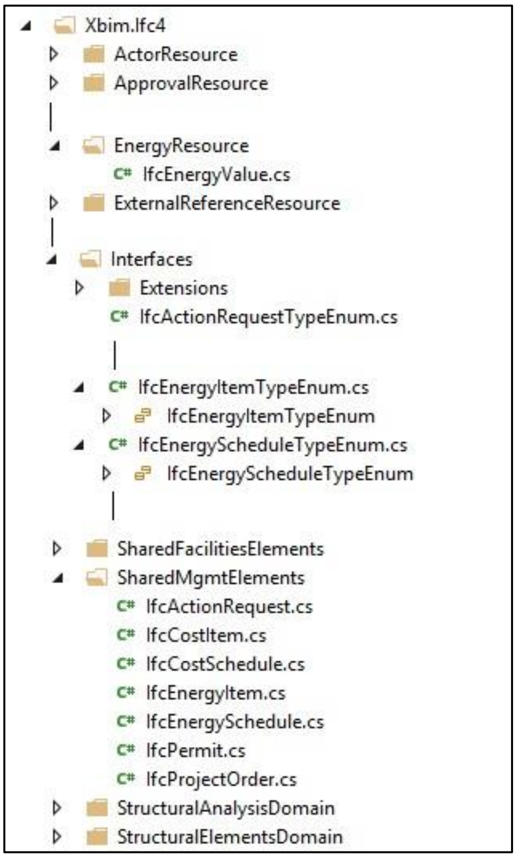


Figure 5-17: Modified XBIM.IFC4

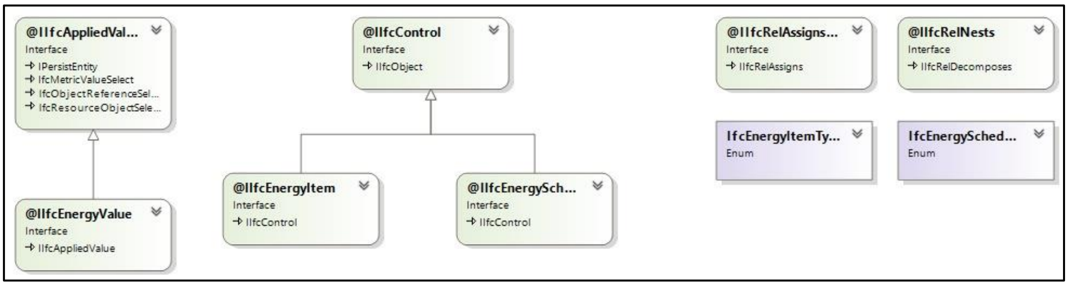


Figure 5-18: Relationship of new classes with the existing ones

5.4.3 Adding IFC functionality to the prototype tool

Once the xBIM tool is modified to incorporate embodied energy entities, the next step is to integrate the actual embodied energy data (calculated with in prototype tool) in the IFC file. Before going into the details of providing this additional functionality to the prototype tool, it is necessary to understand the input and output of the xBIM tool. As discussed in the section 5.4.1, the xBIM tool can create from the scratch or update existing IFC files. In this study, first an IFC file is exported using the built-in capacity of the BIM authoring tool (Revit), then the resulting IFC file is updated with the embodied energy information.

To make the prototype tool capable of reading and modifying the IFC files, a new class of *IfcEnergyInclusion.cs* is added to the prototype. This class is provided with the references of the modified xBIM tool and some code to map the energy data on IFC. The code is responsible to full fill three tasks; (i) to read all the elements in the model with respect to its type such as wall, foundation, roof etc. (ii) Create three *IfcControl*(s) of type *EnergyItem* for each element representing material, transportation and construction energies and (iii) Using *IfcRelNest* to combine all the three energy impacts in one *EnergyItem* which can be called as the total embodied energy for that item. Figure 5-19 below shows the details of the

IfcEnergyInclusion class and its interaction with the prototype tool as well as the xBIM tool.

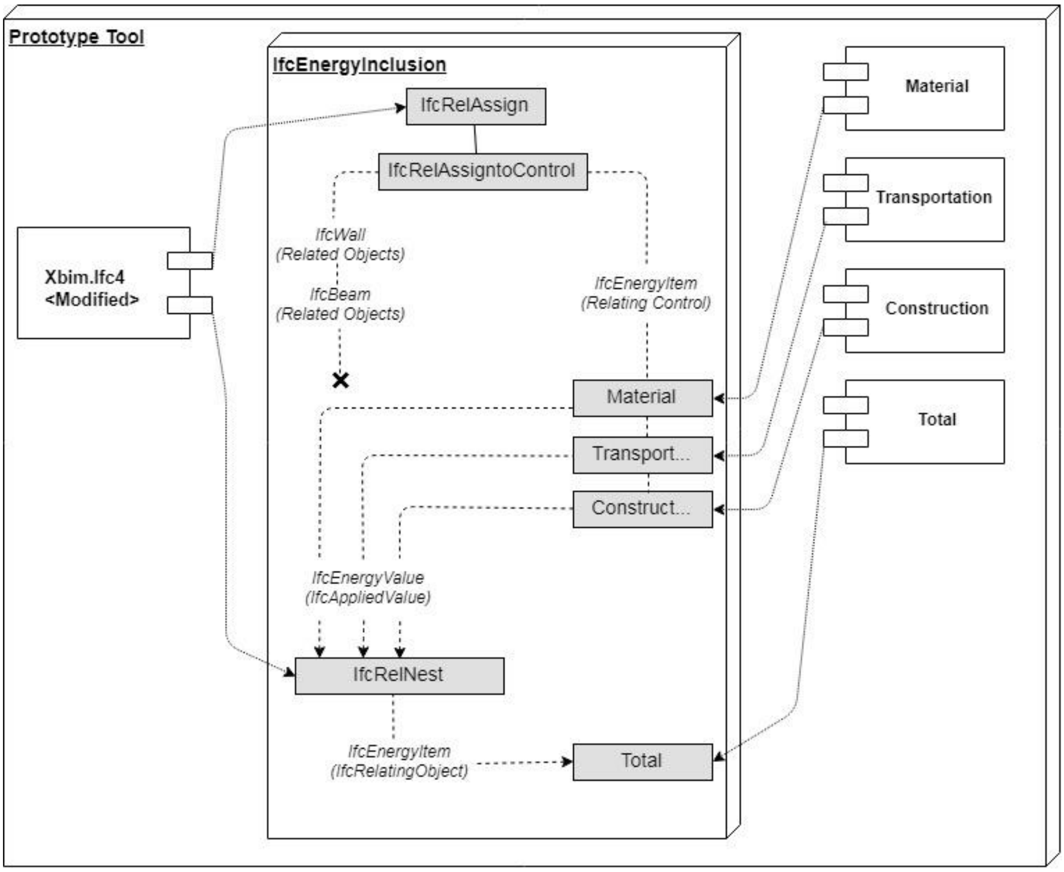


Figure 5-19: Generic interaction diagram of IfcEnergyInclusion

5.4.4 Importing IFC data in the tool

Once the exported file is embedded with the energy data, it can also be imported back in the tool. This is important because when different projects are analyzed with the tool, this acts as storage mechanism for all the embodied energy related results. The main screen of the tool is shown in 5-20, once the IFC file is imported through the interface, rest of the tool is populated with the IFC data.

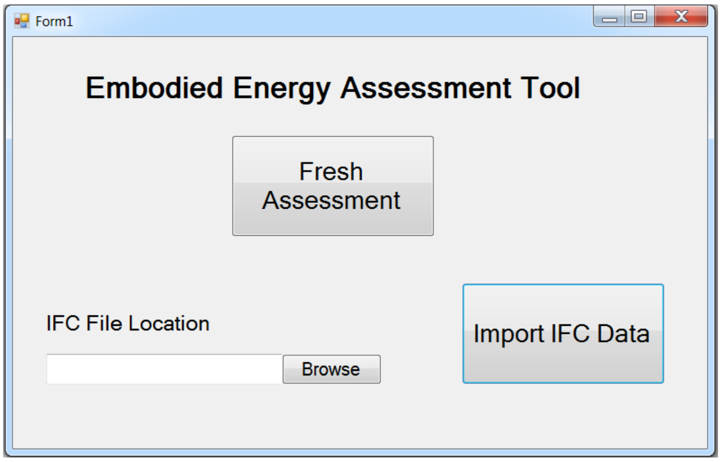


Figure 5-20: Main interface of the prototype tool

5.5 Summary

This chapter explains the working of the prototype tool. The tool comprises of four tabs (Material, Transportation, Construction and Total Energy), each tab is automatically populated with respective data from the BIM model as well as the external databases. The tool also has functionality to simulate various alternatives to provide optimum results for transportation and construction energy. Finally, xBIM toolkit is modified to export the results in IFC format.

CHAPTER 6: CASE STUDY

To implement the framework a case study is conducted to check the applicability of the framework through the developed tool. For the data collection two approaches were implemented: initially 2D construction drawings were procured to develop the BIM model and secondly a questionnaire was drafted and handed over to the construction manager which enquired about the material transportation and construction details of the project. The case study presented here is about a Sangtian island biological industrial park phase II building in China. The project is a cast-in-situ concrete frame structure. There are four floors and the total height is 27.5 m. Figure 6-1 and Figure 6-2 show the 2D east and north section of the building respectively. A BIM model is created using Revit, as shown in Figure 6-3.

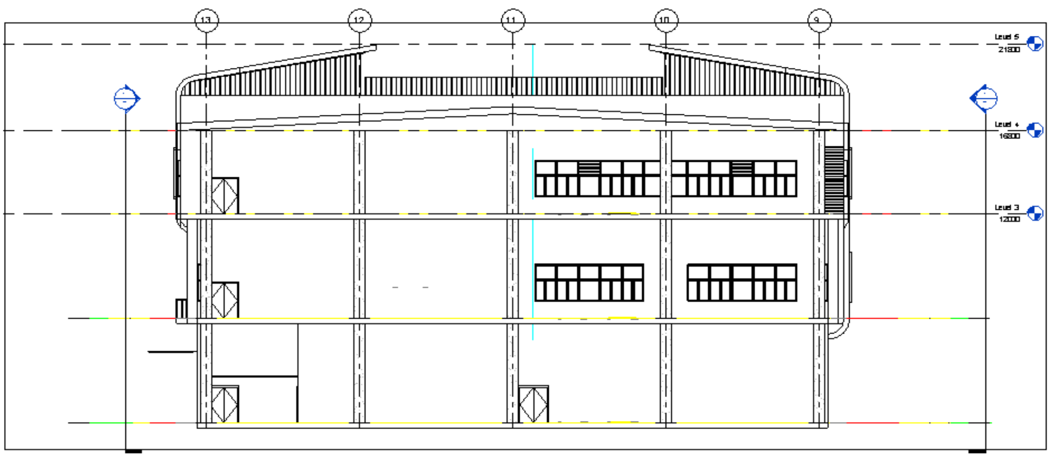


Figure 6-1: East section of the case study building

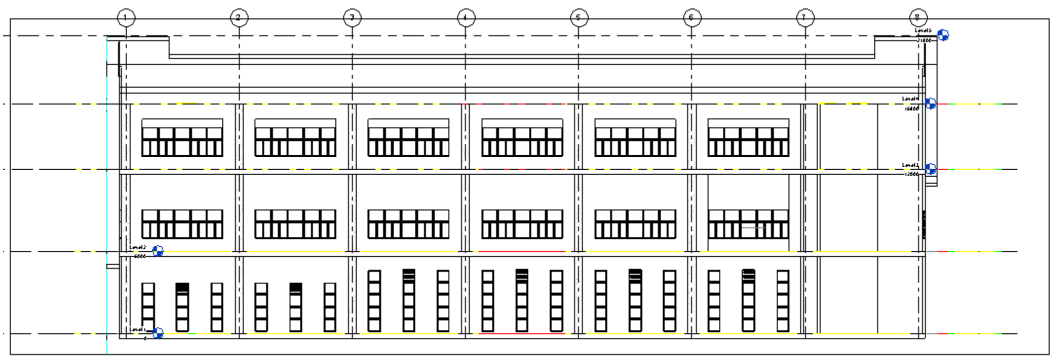


Figure 6-2: North section of the case study building

Once the duly filled questionnaire was received, the designated databases were populated with data such as material supplier distances from site, construction equipment efficiency, etc. Finally, the tool was launched and the material, transportation and construction energy was estimated.

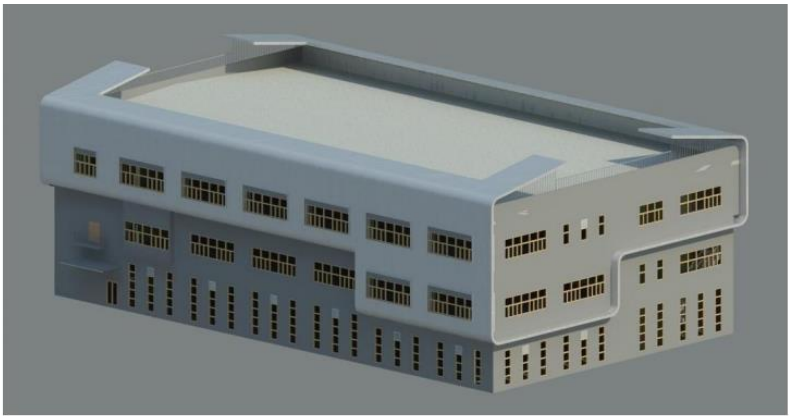


Figure 6-3: BIM model for the case project

6.1 Data Collection

In the first phase data was collected about the earthmoving activity. Table 6-1 presents general conditions of the site and average productivity for excavation. Table 6-2 gives the details about zoning for excavation as well as the start finish times.

Table 6-1: Site conditions and average productivity for Excavation

Data Collection for Excavation		
Project/Location	**Temperature**	**Humidity**
SIBIP Phase II	32℃	Change with Time (Generally, above 90%)

Daily Work Time	**Soil Type**	**No. of Teams**
8a.m~11a.m/ 12a.m~5p.m (generally 8 hours/day)	Clay	1
Depth	**Dilation Co.**	**Total Excavation Volume**
4.5 m	1.15	15000 m^3
Team Productivity	**Excavator per team**	**Labor per team**
50 ~ 60 m^3/hour	1 normal + 1 helper	4
Distance from site to dump	**Preparation T need before Excavation**	**Normal trucks need per day**
25 Km	2~3 days	4

Table 6-2: Data Collection for Excavation

Excavation Zones					
Zone 1	Area m^2	680	**Zone 2**	Area m^2	760
	Volume of soil m^3	3519		Volume of soil m^3	3933
	Start	13 May		Start	23 May
	Finish	23 May		Finish	3 June
Zone 3	Area m^2	735	**Zone 4**	Area m^2	720
	Volume of soil m^3	3803.5		Volume of soil m^3	3726
	Start	4 June		Start	13 June
	Finish	13 June		Finish	22 Jun

In the next phase data was collected about different material manufactories, warehouses and machine leasing companies for the estimation of transportation distances. Once the locations were determined, there distances were calculated using Baidu online maps [83]. Figure 6-4 shows the Suzhou Jincheng Concrete mixing plant, and the distance from the site is 30.9 km.

Figure 6-4: Suzhou Jincheng Concrete mixing plant

The location of the Suzhou Shengyi Reinforcement yard is shown in Figure 6-5, which is at a distance of 34.5 km.

Figure 6-5: Suzhou Shengyi Reinforcement yard

Aluminium was procured from Suzhou Yuancheng aluminum factory as shown in Figure 6-6 and the distance is 38.6 km.

Figure 6-6: Suzhou Yuancheng Aluminum factory

In addition, Figure 6-7 shows the Suzhou Gelin windows and doors processing factory, the distance to the construction site is 40.2 km.

Figure 6-7: Suzhou Gelin windows and doors processing factory

Similarly, Figure 6-8 shows Suzhou Chenguang Steel Template Rental Stations, and the distance to the construction site is 35.6 km

Figure 6-8: Suzhou Chenguang steel template rental stations

Figure 6-9 shows Suzhou Taifeng Scaffold Rental Stations, and the distance to the construction site is 45.5km.

Figure 6-9: Suzhou Taifeng Scaffold Rental Stations

And finally, Figure 6-10 shows the location of the Suzhou Jincheng Construction Machinery Equipment Leasing Co., Ltd., the distance to the construction site is 26.5 km.

Figure 6-10: Suzhou Jincheng Construction Machinery Equipment Leasing Co. Ltd.

In the third phase data was collected for the indirect quantities, which need to be estimated outside the BIM authoring tool. First a ratio was probed for the formwork contact area for different building elements. Table 6-3 gives a ratio of the formwork contact area per cubic meter for different building elements.

Table 6-3: The ratio of the formwork contact area

Item	Unit	Contact area of formwork (m^2)
column	m^3	8.725
beam	m^3	6.606
slab	m^3	4.901
foundation	m^3	2.197
wall	m^3	5.74

As for the scaffolding, the total length, width and height was found to be 62.8m, 37.6m and 30.547m, respectively. Full hall steel tubular scaffold is used in the project as shown in Figure 6-11. The scaffold used in the construction has a diameter of 48mm and the wall thickness of 3.24 mm, with an average weight of

3.97 kg per meter. In this case, the total length needed for the project is 43726.014 m which is based on the length of cross-bar, pole and scissors, and the subsequently the weight of the scaffold is estimated to be 173.592t.

Figure 6-11: Full hall steel tubular scaffold

In the fourth phase, information about the transportation vehicles was collected. The prime focus was on the fuel consumption of different types of trucks and the lower heating value of the diesel. The lower heating value of the diesel is 42.7 MJ/L and the fuel consumption is based on the type of the truck. As for the concrete mixer truck, the typical capacity used in construction site is 12 m^3 which is shown in Figure 6-12. For this type of the mixer, the fuel consumption is 0.5 L/km.

Figure 6-12: Concrete mixer truck

For hauling other materials, it is important to note the nature of different materials dictate the utility of hauling vehicle in a particular manner. To explain it further, consider an 8- ton loading truck as shown in Figure 6-13. This same truck when loading a material like reinforcing steel, the consideration of its maximum weight carrying capacity is enough for the analysis. However, when material like formwork is considered, the capacity in terms of volume becomes crucial as the weight may be below the capacity but there may not be enough space. So, for the formwork the capacity of the truck was found to be 55.2 m^3

Figure 6-13: Loading Truck (8 ton)

In the fifth phase, data was collected for the transportation of staff and heavy equipment. Most of the workers were living within the construction site, so just managerial staff was considered for transportation. To simplify, an average value of 25 KM was agreed upon and the vehicles driven by this staff was found to be gasoline fuelled sedan. Twenty numbers of staff were allocated on the project, who worked 8 hours a day. The total duration for the project is one year which is 365 days. Considering the national statutory holidays, the round trips for the project were found to be 333. And total trips were estimated to be 666. The fuel consumption of the general sedan is 10 L/100km, and the lower heating value of the gasoline is 44 MJ/L.

As far as the transportation of machinery is concerned, all of the machinery is divided into four parts: earthwork machine, hoisting machinery, rebar machinery and concrete machinery as shown in Table 6-4. All the machines were leased from a single machine's leasing company as mentioned previously and were transported on the site by using a flat bed truck as shown in Figure 6-14.

Table 6-4: List of heavy equipment.

Application	Equipment Names
Earthwork	Excavator
	Loader
	Bulldozer
	Scraper
	Grader
Hoisting machinery	Crawler crane
	Tower crane
	Winding engine
	Rebar machinery
Rebar machinery	Steel coolers
	Steel cold-drawing machine
	Steel-bar cutter
	Steel-bar straightener
	Steel-bar bender
	Butt welder
	Spot welder
Concrete Machinery	Concrete mixer
	Concrete pump
	Concrete vibrator
	Shotcrete machine

Figure 6-14: Machinery transported to site by using flat bed truck

In the last phase, the data was collected about the construction equipment productivity and power rating. Table 6-5 lists the team-based productivity of various construction equipments as well as their power ratings. This data is adopted from the Chinese construction handbook [77]. According to the Table, the workload per machine per team and rated power have been referred. The second column is the workload per machine per team which points to the total workload in 8 hours.

Table 6-5: Summary of the typical construction equipment

Machinery	**Workload per machine per team**	**Rated power (KW)**
Excavator	350 m^3	58.8
Bulldozer	300 m^3	74.6
Steel-Bar Cutter	16 t	5.5
Steel-Bar Straightener	3 t	15
Steel-Bar Bender	8 t	4
Butt Welder	7 t	25
Concrete Mixer	100 m^3	10
Concrete Pump	64 m^3	97
Immersion Vibrator	24 m^3	2
Flat-Plate Vibrator	56 m^3	1.5
Crawler Crane	10 t	65

6.2 Prototype Tool Based Assessment Results

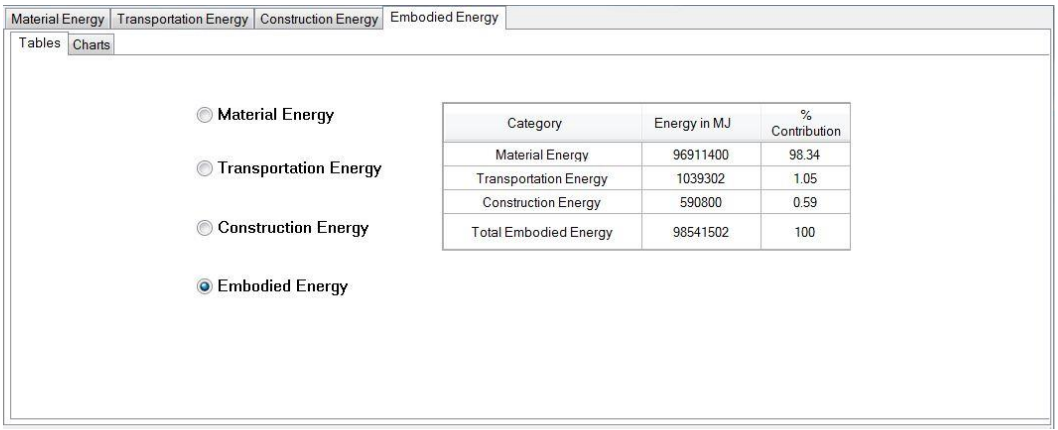


Figure 6-15: Embodied Energy distribution

The embodied energy assessment results from the tool are in the form of tables and charts. Figure 6-15 represents the results of the embodied energy broken down in material embodied energy, transportation energy and construction energy. Material energy constitutes about 98% of the embodied energy content of the building.

Figure 6-16 shows building element-wise material embodied energy distribution. The foundation contributes highly in terms of material energy distribution, while columns are found to be the lowest contributors.

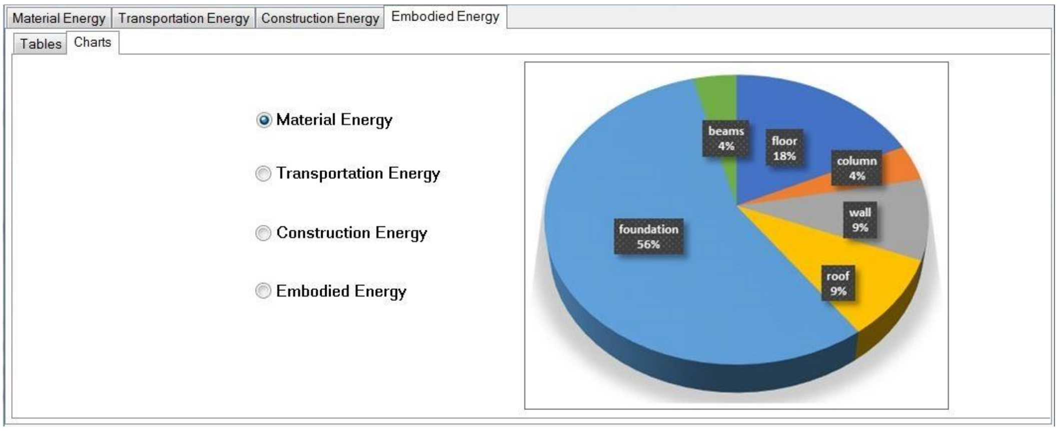


Figure 6-16: Material Energy Contribution of each element

Since the building model comprises the concrete frame with reinforced concrete members, the material transportation included primarily two materials: concrete and steel. The material transportation energy contribution of both materials is presented in Figure 6-17.

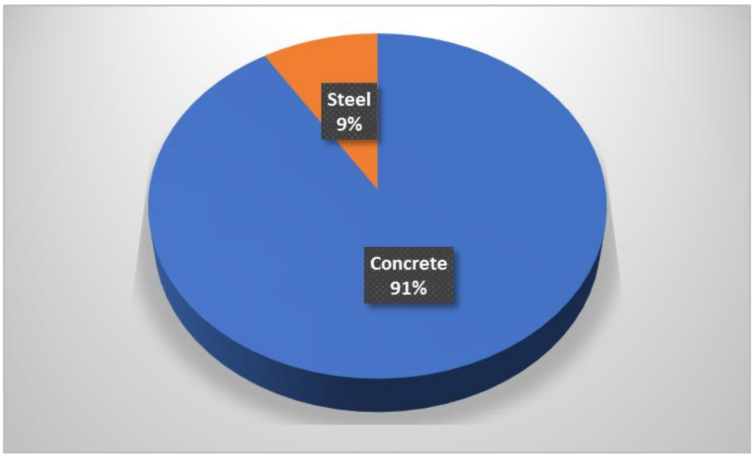


Figure 6-17: Transportation Energy Contribution of each material

For construction energy, the tool breaks down the energy consumption in terms of construction equipment energy consumption. Figure 6-18 shows a breakdown of construction energy utilization in terms of the construction equipment. Concrete pump dominates the construction energy consumption with more than sixty percent utilization.

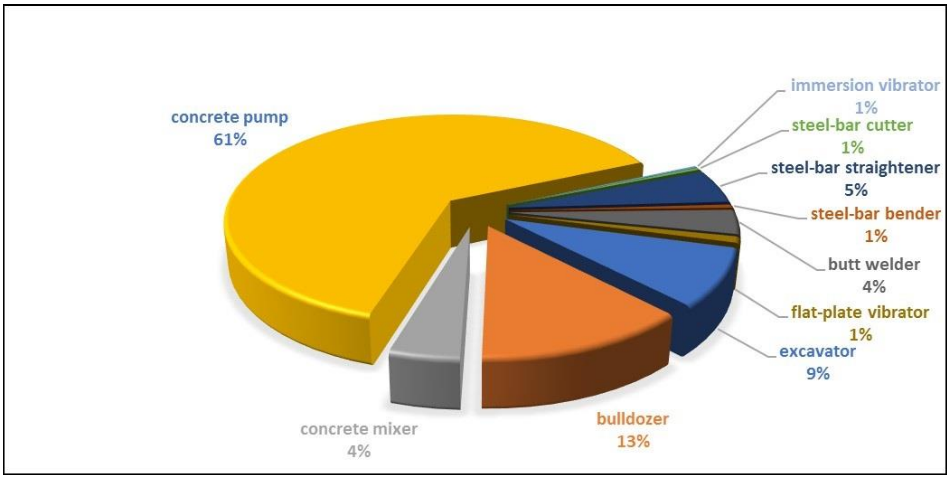


Figure 6-18: Construction Energy Contribution of each equipment

6.2.1 Validation of Results

The validation of the results is done through detailed manual calculations. First the calculations for estimating material energy are performed using equations 3.2 and 3.3. Table 6-6 and 6-7 present the calculation of material energy for concrete and steel respectively.

Table 6-6: Element wise material energy calculation for concrete

Concrete				
Element	Quantity (m^3)	EE coefficient (MJ/kg)	Density (Kg/m^3)	EE (MJ)
Floor	1917.3	1.95	2500	9346837.5
column	275.62	3.5	2500	2411675
Wall	970.01	1.95	2500	4728798.75
Roof	925.54	1.95	2500	4512007.5
foundation	4200	3.5	2500	36750000
beams	310.05	3.5	2500	2712937.5
Total				60462256.25

Table 6-7: Element wise material energy calculation for steel

Steel				
Element	Quantity (m^3)	EE coefficient (MJ/kg)	Density (Kg/m^3)	EE (MJ)
floor	47.9325	21.6	7850	8127434.7
column	6.8905	21.6	7850	1168353.18
wall	24.25025	21.6	7850	4111872.39
roof	23.1385	21.6	7850	3923364.06
foundation	105	21.6	7850	17803800
beams	7.75125	21.6	7850	1314301.95
Total				36449126.28

Table 6-8 presents element wise distribution of the material energy; the results are similar to the results obtained by the tool as shown in Figure 6-5. Transportation energy is estimated by using equations 3.4 and 3.5; Table 6-4 presents the material wise distribution of the transportation energy. The percentage contributions of materials are similar to the results obtained in Figure 6-6. Similarly, the construction energy is estimated using equation 3.6, 3.7 and 3.8. Since different set of equipment are used to perform different work packages, the calculations are done separately for different packages. Table 6-9 to Table 6-15 present the calculations for different work packages consisting of foundations, columns, beams, walls, floors and roofs.

Table 6-8: Element wise material energy distribution

Element	**Embodied Energy in Concrete**	**Embodied Energy in Steel**	**Total material embodied energy**	**Percentage Contribution**
floor	9346837.5	8127434.7	17474272.2	18
column	2411675	1168353.18	3580028.18	3.7
wall	4728798.75	4111872.39	8840671.14	9.1
roof	4512007.5	3923364.06	8435371.56	8.7
foundation	36750000	17803800	54553800	56.3
beams	2712937.5	1314301.95	4027239.45	4.2
Total	60462256.25	36449126.28	**96911382.53**	100

Table 6-9: Material wise transportation energy distribution

Material	**Qty. (m^3 or t)**	**Cpty. (m^3 or t)**	**Rnd. Trips**	**Dist. (Km)**	**Fuel cons. (L/Km)**	**Lower Heat. Value (MJ/L)**	**Trans. Energy (MJ)**	**% Contr.**
Concrete	8598.52 m^3	12 m^3	717	30.9	0.5	42.7	946031.31	91
Steel	1687.45955 t	8 t	211	34.5	0.15	42.7	93250.395	9
							1039281.70	**100**

Table 6-10: Construction energy estimation for foundations

Foundation							
Equipment	**Qty.**	**Workload per machine per day (m^3)**	**Working hours**	**Rated power (KW)**	**Electric work (KW.h)**	**KW.h to MJ conversion factor**	**Energy (MJ)**
excavator	10500	350	240	58.8	14112	3.6	50803.2
bulldozer	10500	300	280	74.6	20888	3.6	75196.8
concrete mixer	4200	100	336	10	3360	3.6	12096
concrete pump	4200	64	525	97	50925	3.6	183330
flat-plate vibrator	4200	56	600	1.5	900	3.6	3240
steel-bar cutter	105	16	52.5	5.5	288.75	3.6	1039.5
steel-bar straightener	105	3	280	15	4200	3.6	15120
steel-bar bender	105	8	105	4	420	3.6	1512
butt welder	105	7	120	25	3000	3.6	10800
Total							353137.5

Table 6-11: Construction energy estimation for column

Column							
Equipment	Qty.	Workload per machine per day (m^3)	Working hours	Rated power (KW)	Electric work (KW.h)	Kw.h to MJ conversion factor	Energy (MJ)
concrete mixer	275.62	100	23	10	230	3.6	828
concrete pump	275.62	64	35	97	3395	3.6	12222
immersion vibrator	275.62	56	40	2	80	3.6	288
steel-bar cutter	2.625	16	2	5.5	11	3.6	39.6
steel-bar straightener	2.625	3	7	15	105	3.6	378
steel-bar bender	2.625	8	3	4	12	3.6	43.2
butt welder	2.625	7	3	25	75	3.6	270
Total							14068.8

Table 6-12: Construction energy estimation for beams

Beams							
Equipment	Qty.	Workload per machine per day (m^3)	Working hours	Rated power (KW)	Electric work (KW.h)	Kw.h to MJ conversion factor	Energy (MJ)
concrete mixer	310.05	100	25	10	250	3.6	900
concrete pump	310.05	64	39	97	3783	3.6	13618.8
immersion vibrator	310.05	56	45	2	90	3.6	324
steel-bar cutter	6.8905	16	4	5.5	22	3.6	79.2
steel-bar straightener	6.8905	3	19	15	285	3.6	1026
steel-bar bender	6.8905	8	7	4	28	3.6	100.8
butt welder	6.8905	7	8	25	200	3.6	720
Total							16768.8

Table 6-13: Construction energy estimation for walls

Walls							
Equipment	Qty.	Workload per machine per day (m^3)	Working hours	Rated power (KW)	Electric work (KW.h)	Kw.h to MJ conversion factor	Energy (MJ)
concrete mixer	970.01	100	78	10	780	3.6	2808
concrete pump	970.01	64	122	97	11834	3.6	42602.4
immersion vibrator	970.01	56	139	2	278	3.6	1000.8
steel-bar cutter	24.25025	16	13	5.5	71.5	3.6	257.4
steel-bar straightener	24.25025	3	65	15	975	3.6	3510
steel-bar bender	24.25025	8	25	4	100	3.6	360
butt welder	24.25025	7	28	25	700	3.6	2520
Total							53058.6

Table 6-14: Construction energy estimation for roof

Roof							
Equipment	Qty.	Workload per machine per day (m^3)	Working hours	Rated power (KW)	Electric work (KW.h)	Kw.h to MJ conversion factor	Energy (MJ)
concrete mixer	925.54	100	74.0432	10	740.432	3.6	2665.5552
concrete pump	925.54	64	115.6925	97	11222.173	3.6	40399.821
flat-plate vibrator	925.54	56	132.22	1.5	198.33	3.6	713.988
steel-bar cutter	23.1385	16	11.56925	5.5	63.630875	3.6	229.07115
steel-bar straightener	23.1385	3	61.702667	15	925.54	3.6	3331.944
steel-bar bender	23.1385	8	23.1385	4	92.554	3.6	333.1944
butt welder	23.1385	7	26.444	25	661.1	3.6	2379.96
Total							50053.534

Table 6-15: Construction energy estimation for floor

Floor							
Equipment	Qty.	Workload per machine per day (m^3)	Working hours	Rated power (KW)	Electric work (KW.h)	Kw.h to MJ conversion factor	Energy (MJ)
concrete mixer	1917.3	100	153.384	10	1533.84	3.6	5521.824
concrete pump	1917.3	64	239.6625	97	23247.263	3.6	83690.145
flat-plate vibrator	1917.3	56	273.9	1.5	410.85	3.6	1479.06
steel-bar cutter	47.9325	16	23.96625	5.5	131.81438	3.6	474.53175
steel-bar straightener	47.9325	3	127.82	15	1917.3	3.6	6902.28
steel-bar bender	47.9325	8	47.9325	4	191.73	3.6	690.228
butt welder	47.9325	7	54.78	25	1369.5	3.6	4930.2
Total							103688.27

Table 6-16 presents the equipment wise construction energy estimations, while Table 6-17 presents the contributions for different construction equipment towards construction energy. The results found in Table 6-12 are similar to that obtained by the tool as shown in Figure 6-4.

Table 6-16: Equipment wise construction energy estimations

Equipment	Foundation	Column	Beams	Wall	Roof	Floor	Total
excavator	50803.2	0	0	0	0	0	50803.2
bulldozer	75196.8	0	0	0	0	0	75196.8
concrete mixer	12096	828	900	2808	2665.5552	5521.824	24819.38
concrete pump	183330	12222	13618.8	42602.4	40399.821	83690.145	375863.2
immersion vibrator	0	288	324	1000.8	0	0	1612.8
steel-bar cutter	1039.5	39.6	79.2	257.4	229.07115	474.53175	2119.303
steel-bar straightener	15120	378	1026	3510	3331.944	6902.28	30268.22
steel-bar bender	1512	43.2	100.8	360	333.1944	690.228	3039.422
butt welder	10800	270	720	2520	2379.96	4930.2	21620.16
flat-plate vibrator	3240	0	0	0	713.988	1479.06	5433.048
Total							590775.5

Table 6-17: Contributions for different construction equipment towards construction energy.

Equipment	Percentage Contribution
excavator	8.6
bulldozer	12.73
concrete mixer	4.2
concrete pump	63.61
immersion vibrator	0.28
steel-bar cutter	0.36
steel-bar straightener	5.12
steel-bar bender	0.52
butt welder	3.66
flat-plate vibrator	0.92
Total	100

The results obtained from the tool match the results of the manual calculation which validates the accuracy of the workings of the tool.

6.2.2 Additional Assessment

6.2.2.1 Material Embodied Energy

After initial assessment and validation, the tool was used to further investigate about the embodied energy content for other materials. The glass used in windows is double layers and filled with air except the blind which is made with aluminium alloy. The thickness ratio of the glass-air-glass is 6-8-6 mm. The estimation of embodied energy for glass is presented in Table 6-18.

Table 6-18: Embodied energy estimation for glass

Glass				
Type	Quantity (m^3)	EE coefficient (MJ/kg)	Density (kg/m^3)	EE (MJ)
Window	36.552	23.5	2500	2147430

Similarly, aluminium is used for blinds, doors, handrail and facade. The quantity of blind and door can be retrieved from software, but the quantity of handrail and facade has been calculated manually. The embodied energy for aluminium is estimated as Table 6-19.

Table 6-19: Embodied energy estimation for aluminium

Aluminium				
Type	**Quantity (m^3)**	**EE coefficient (MJ/kg)**	**Density (kg/m^3)**	**EE (MJ)**
Blind	0.91	155	2700	380835
Door	4.77	155	2700	1996245
Handrail	0.14	155	2700	58590
Façade	3.74	155	2700	1565190
Total				4000860

The total material embodied energy including the additional materials is shown in Table 6-20 which sums up to be 103059672 MJ.

Table 6-20: Total material embodied energy

Material	**EE**	**EE percentage**

Concrete	60462256	58.66%
Steel	36449126	35.36%
Aluminium	4000860	3.90%
Glass	2147430	2.08%
Total	103059672	100%

6.2.2.2 Transportation Energy

As per the data collected in the second and third phase, the transportation energies for aluminum, glass, formwork and scaffold are also estimated. Using the ratios given in Table 6-3, and the material quantity obtained from the Revit, the area of formwork is estimated as shown in Table 6-21. The contact area quantity of the formwork is 5117.933 m^2, the rib height of the formwork is 55mm. In this case, the volume of the formwork is 281.486 m3. So the Table 6-22 presents the estimation of additional material transportation embodied energy.

Table 6-21: Estimation of total formwork used

Item	**Quantity of component (m^3)**	**Contact area of formwork (m^2)**	**Quantity (m^2)**
column	91.87	8.725	801.57
wall	117.1025	5.74	672.168
floor	639.1	4.901	3132.23
beam	77.5	6.606	511.965
Total		5117.933	

Table 6-22: Additional material transportation embodied energy

Material	**Quantity (t)**	**Capacity (t)**	**Round trips**	**Distance (km)**	**Fuel consump (L/km)**	**Lower hearting value (MJ/L)**	**Energy (MJ)**
Aluminum	25.812	8	8	38.6	0.15	42.7	1977.864
Glass	91.38	8	24	40.2	0.15	42.7	6179.544
Formwork	281.486 m3	55.2 m3	10	35.6	0.15	42.7	2280.18
Scaffold	173.592	8	44	45.5	0.15	42.7	12822.81

Furthermore, the estimation of staff transportation energy and machine transportation energy is based on the data collected in the fifth phase and presented in Table 6-23 and 6-24 respectively.

Table 6-23: Staff Transport Energy

Quantity	**Total Trips**	**Distance (km)**	**Fuel consumption (L/km)**	**lower heating value (MJ/L)**	**Energy (MJ)**
20	666	25	0.1	44	1465200

Table 6-24: Machine Transportation Energy

Name	**Total Trips**	**Distance (Km)**	**Fuel Consump. (L/Km)**	**Lower Heating Value (MJ/L)**	**Energy (MJ)**
Excavator	2	26.5	0.28	42.7	633.668
Loader	2	26.5	0.28	42.7	633.668
Bulldozer	2	26.5	0.24	42.7	543.144
Scraper	2	26.5	0.2	42.7	452.62
Grader	2	26.5	0.15	42.7	339.465
Crawler Crane	2	26.5	0.33	42.7	746.823
Tower Crane	2	26.5	0.29	42.7	656.299
Total			4005.687		

Some machines could not be transported with normal sized truck and thus needed to be disassembled and transported in parts. A 10-ton truck has a general length of about 6 meters which is not long enough to load all the rebar machinery and concrete machinery. Table 6-25 shows the estimated transportation energy required for transporting rebar and concrete machinery.

Table 6-25: Special Machine Transportation Energy

Machine	capacity (t)	round	distance (km)	fuel consumption (L/km)	lower hearting value (MJ/L)	Energy (MJ)
Rebar and concrete machinery	10	6	26.5	0.17	42.7	1154.181

6.2.2.3 On-site Transportation Energy

In contrast to the transportation of material, machinery and staff from some other origin to the construction site, there is some internal transportation known as on-site transportation. On-site transportation can be divided into vertical transportation and horizontal transportation, vertical transportation includes moving material or machine from a point on the ground to an elevated platform and vice versa. And the horizontal transportation, includes the transportation from material store yard to construction site. For this project, the average distance from the material store yard to site is 50m. Table 6-26 shows the energy consumption of on-site horizontal transportation.

Table 6-26 On-site horizontal transportation.

Material (Platform Truck)	**quantity (t)**	**capacity (t)**	**round**	**distance (km)**	**fuel consump. (L/km)**	**lower hearting value (MJ/L)**	**Energy (MJ)**
formwork	281.486 m3	55.2 m3	10	0.1	0.15	42.7	6.405
rebar	562.845	8	142	0.1	0.15	42.7	90.951
Total							97.356

In addition, Table 6-27 shows the energy consumption in on-site vertical transportation. The quantity of the formwork for vertical transportation is not the same as the horizontal transportation this is because of the reusability of formwork. According to the steel formwork weighing scale, the weight of each 0.42 m^2 per piece formwork is 16.2 kg, the total weight of formwork has been calculated to be

197.405t. However, the same formwork is transported thrice vertically for second floor, third floor and roof so the quantity is considered to be 789.623. But the quantity of rebar would remain same as before.

Table 6-27: Energy consumption in on-site vertical transportation.

Material (Crawler Crane)	**Qty (t)**	**Workload per machine per team (t)**	**Working hours (h)**	**Rated power (KW)**	**Electric work (KW.h)**	**Energy (MJ)**
formwork	789.623	10	631.70	65	41060.5	147817.8
rebar	417.91	10	334.33	65	21731.32	78232.75
Total						226050.55

6.2.2.4 Temporary Structure Embodied Energy

Consequently, the embodied energy content of the temporary structures is also estimated. As for the formwork and scaffold work, based on previous calculations, the quantity is found out to be 197.405t and 173.592t. In addition, according to the embodied energy in the material and the service life of the material. The energy consumption in formwork and scaffold work has been calculated in Table 6-28.

Table 6-28: Total energy consumption for formwork and scaffold work.

Type	**Quantity (t)**	**EE coefficient (MJ/Kg)**	**EE (MJ)**
formwork	197.405	32	6316960
scaffold	173.592	24.9	4322440.8
Total			10364900.8

The service life for a good protected formwork and scaffold is 8 years. Since the project duration is one year, the embodied energy consumption in formwork and scaffold work is 1295612.5 MJ. The temporary space energy overhead for this project includes the electricity and gas consumption of office area and the living area for all the workers related with the project. According to the temporary work construction specifications, the floor space for a four-floor construction project of the office area and the living area are 100 m^2 and 300 m^2. In addition, the height of the worker steel room should not exceed two floors. In this case, the total area for office area becomes 200 m^2 and 600 m^2 for living area.

As mentioned above that there are 333 working days involved in this project, and since during the month of March, April, October and November there is no need for air conditioning, the total need for air condition would be 233 days. Here an assumption is made that both the office area and living area will be used 10 hours a day. The power consumption in temporary work has been estimated as shown in Table 6-29.

Table 6-29: Power consumption in temporary work

Item	unit power consumption per day (KW.h/m^2)	Area (m^2)	Electric work (KW.h)	Duration	Energy (MJ)
lighting	0.05	800	40.00	333	47952
air condition	0.133	800	106.67	233	89472
computer	1.5	40	60.00	333	71928
Total					209352

In addition to this energy overhead, workers have to cook using natural gas. Considering a rough estimate of gas consumption of such a facility, combined with the lower heating value of the natural gas, the energy consumption has been estimated as shown in 6-30.

Table 6-30: Gas consumption in temporary work

Fuel	**Quantity (m3)**	**Lower heating value (MJ/m3)**	**Energy (MJ)**
gas	12000	6	72000

Combined with the energy consumption in temporary work, total energy consumption in supporting construction processes has been estimated to be 10457835 MJ.

6.3 Simulation of Construction and Transportation Energy

6.3.1 Simulation Model Results of Transportation Energy

The simulation model for the transportation is already discussed in section 5.2. The coefficients C1, C2 and C3 were calculated to reflect the energy consumption per unit time for loader and truck (loaded and unloaded case). Table 6-31 shows the data considerations for the transportation of steel rebars using base scenario. The second column gives duration for different activites and the forth column gives coefficient (C1) for loaders, C2 for loaded trucks and C3 for unloaded trucks.

Table 6-31: Base Scenario considerations for Transportation of Steel Rebars

Activity	**Duration (mins)**	**Resource/Entity**	**Resource Energy (MJ/min)/Quantity**
Load	30	Loader	4.3554 MJ/min
Travel	60	Trucks (10ton)	3.6295 MJ/min
Unload	25	Loader	4.3554 MJ/min
Travel Back	30	Trucks (10ton)	7.259 MJ/min

The total time taken to transport 563 tons of steel is 2175 minutes while the energy consumed is equivalent to 39198MJ. In order to optimize the duration and energy consumption, three variants in the capacity of the trucks (8 ton, 10 ton and 12 ton) with different number of trucks are experimented. The results are tabulated in Table 6-32 below. The first column represents capacity of trucks, and the other columns gives duration and energy consumption with respect to number of trucks.

Table 6-32: Duration (min) and energy consumption (MJ) with variants in truck capacity and number

Capacity/Nos	3 trucks	4 trucks	5 trucks	6 trucks	7 trucks	8 trucks
8tons	3265 min 41697 MJ	2480 min 41806 MJ	2025 min 42107 MJ	1885 min 42107 MJ	1885 min 42107 MJ	1885 min 42107 MJ
10tons	2815 min 38741 MJ	2175 min 39198 MJ	1825 min 39198 MJ	1825 min 39198 MJ	1825 min 39198 MJ	1825 min 39198 MJ
12tons	2600 min 37798 MJ	2000 min 37972 MJ	2000 min 37972 MJ	2000 min 37972 MJ	--	--

To better understand and analyse these results, two strategies are adopted. The first strategy encompasses the comparison of these results to identify the optimum scenario. The second strategy is focused on the investigation of these results by observing the auxiliary results of the simulation such as production rates and utilization of resources. Such investigations help to identify the bottle necks as well as provides guidance to improve the overall processes.

Figure 6-19 below shows a comparison of transportation time for different truck sizes as well as a varying number of trucks. It can be observed that the red line depicting the 10-ton truck achieves the lowest point at 5 number of trucks which is actually the minimum (optimum) time of 1825 minutes. However, the green line depicting the 12-ton truck achieves its minimum of 2000 minutes at only 4 trucks. The blue line depicting the 8-ton truck remains the worst choice with respect to time as it achieves its minimum of 1885 minutes at 6 number of trucks which is still higher than the minimum of 10-ton truck.

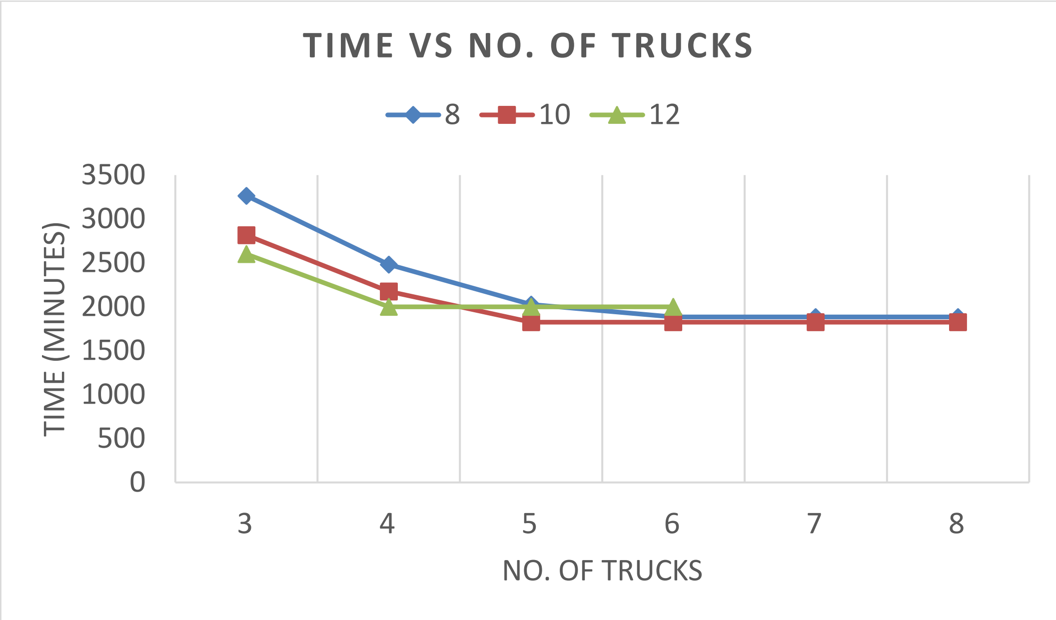


Figure 6-19: Impact of different truck sizes and number of trucks on transportation time.

Figure 6-20 below shows a comparison of transportation energy for different truck sizes as well as a varying number of trucks. It can be observed that there is little impact on the overall energy by varying the number of trucks. However, the 12-ton truck maintains the lowest energy consumption with an optimum (minimum) value of 37798 Mega-Joule at 3 number of trucks. Even for 4 number of trucks which has the minimum time in the 12-ton truck, the transportation energy is just above the minimum with a value of 37972 Mega-Joule. The 10-ton and 8-ton gives truck an average value of 39000 and 42000 Mega-Joule respectively. Therefore, the best choice after considering both time and energy would be to select 4 numbers of 12-ton trucks.

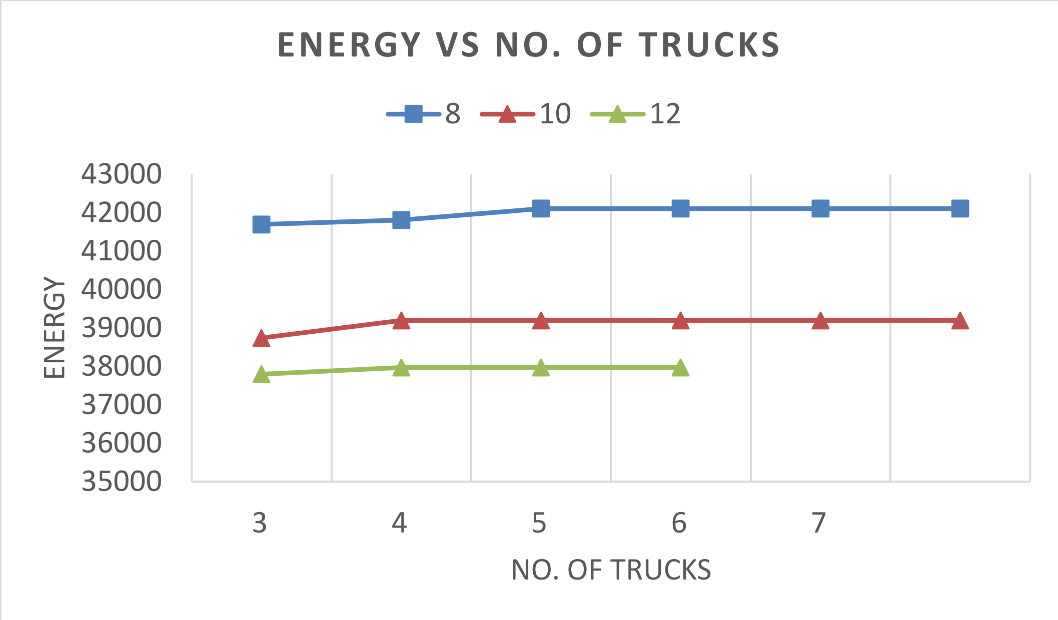


Figure 6-20: Impact of different truck sizes and number of trucks on transportation energy

Now, to investigate these results as to why such results were obtained, the auxiliary results of the model must be taken into consideration. The first thing to note is that the 12-ton truck achieved its optimum time with only 4 trucks while the other variants still bettered their time by increasing the number of trucks. By observing the graphs in Figure 6-21, it can be seen that the graph of 12-ton variant for 4 number of trucks smoothened out while the graphs of other variants for the same number of trucks smoothened out later with respect to the number of trucks. This lack of variation depicts constant value which in turn indicates the achievement of maximum production rate. This can be verified by observing Figure 6-22, that the production rate remained constant for all the variants once their production graph smoothened.

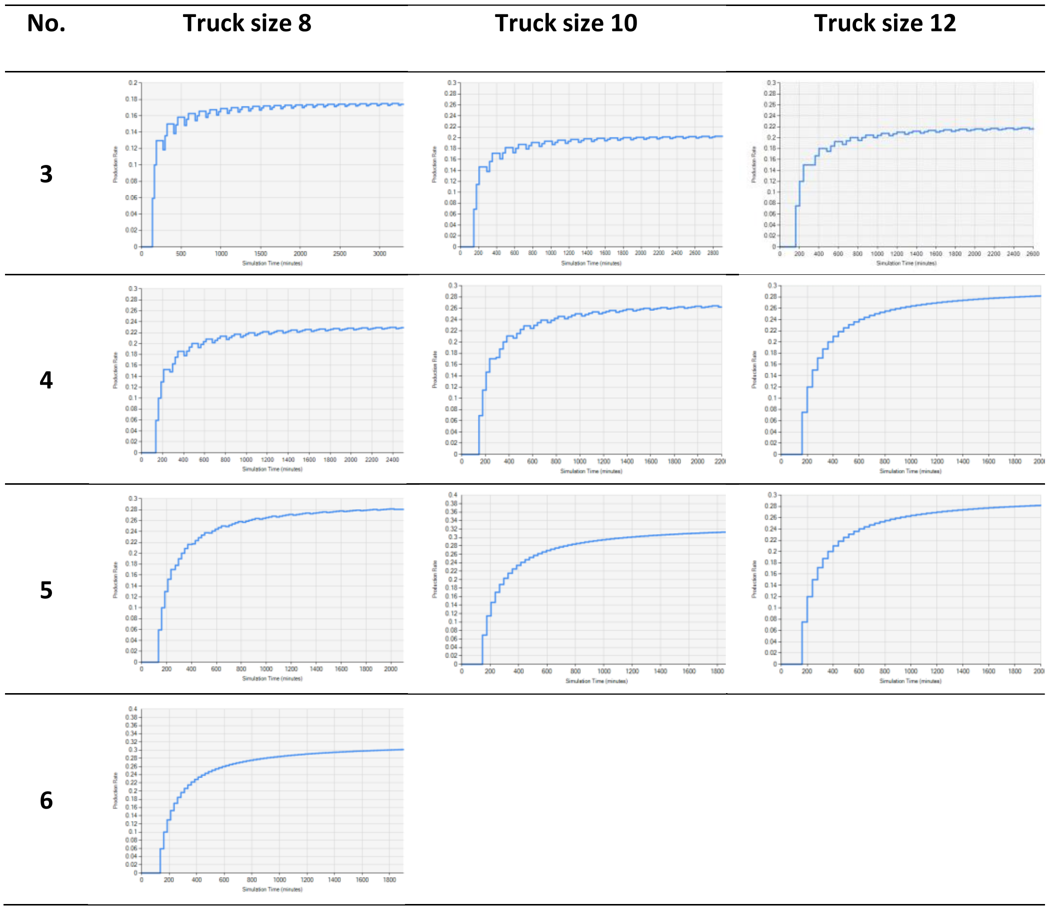


Figure 6-21: Production rates for different truck sized and number of trucks

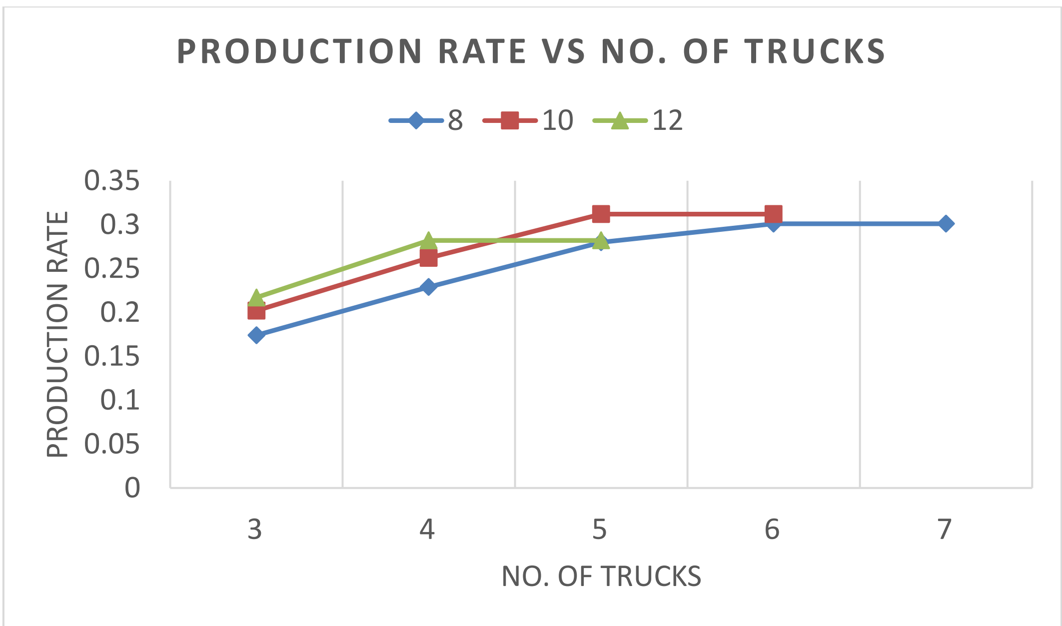


Figure 6-22: Comparison of production rates of different truck sizes and number of trucks

Even though the 12-ton truck reached its optimum with only four trucks, but even by increasing the number of trucks the simulation times could not be bettered. Another aspect that impacted this production rate is the utilization of the resources. By observing Figure 6-23, it can be seen that the utilization of the loader resource reaches 100% with only four of the 12-ton truck variant. So, the bottle neck for this scenario happens to be the capacity or number of loader resource and not the truck itself.

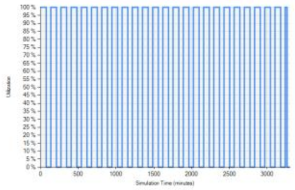
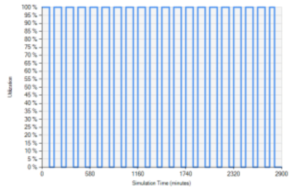
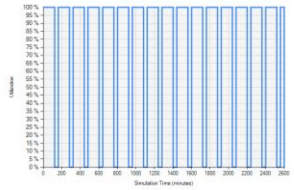
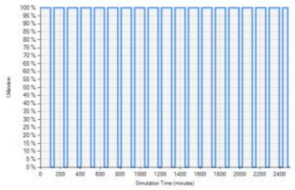
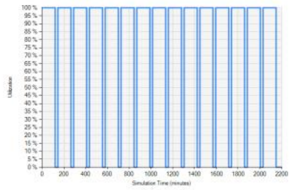
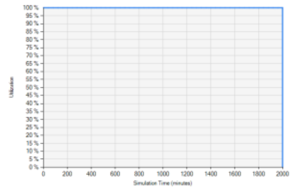
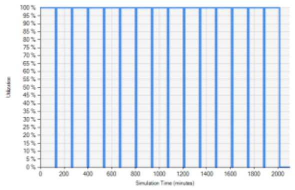
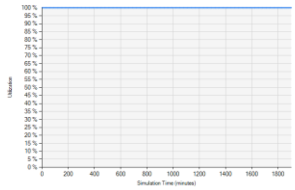
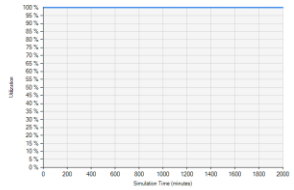
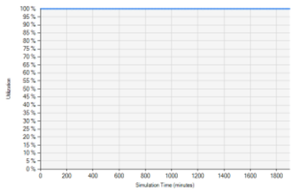
No.	Truck size 8	Truck size 10	Truck size 12
3			
4			
5			
6			

Figure 6-23: Utilization graphs of loader resource

To further investigate utilization of the loader, a comparison is made in Figure 6-24. It can be seen that for all the variants that as soon as the loader resource achieves 100% utilization, production rates become maximum and the minimum times are achieved at that point. To be sure of this observation, the only other resource in the model i.e. unloader is also investigated. Figure 6-25 presents the utilization of the unloader resource, none of the variants were found to have achieved 100 % utilization of the unloader resource. Figure 6-26 illustrates a comparison of the utilization of the unloader resource with respect to the all the variants and number of trucks. The graph is found similar to the loader resource utilization but 80% of maximum utilization. These results clearly strengthen the observation that the major bottle neck happens to be the loader resource.

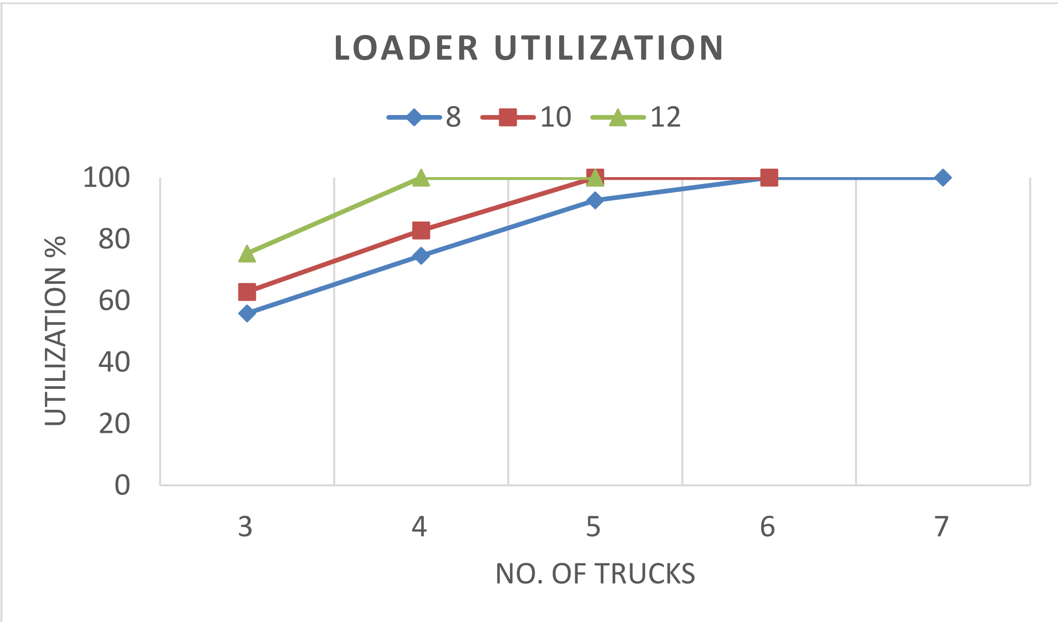


Figure 6-24: Percentage utilization of loader resource for different scenarios

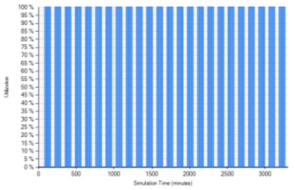
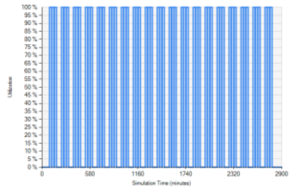
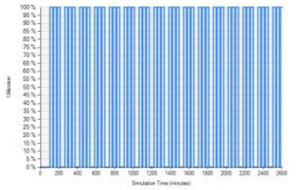
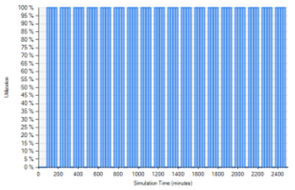
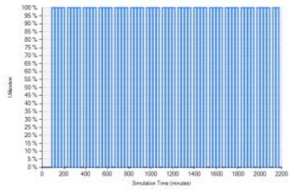
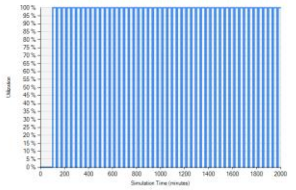
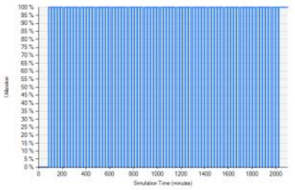
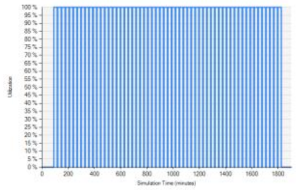
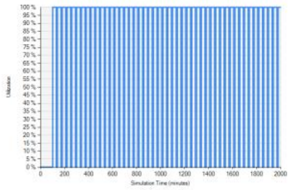
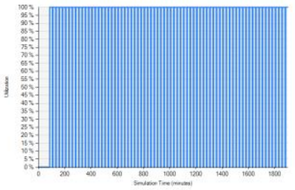
No.	Truck size 8	Truck size 10	Truck size 12
3			
4			
5			
6			

Figure 6-25: Utilization graphs of loader resource

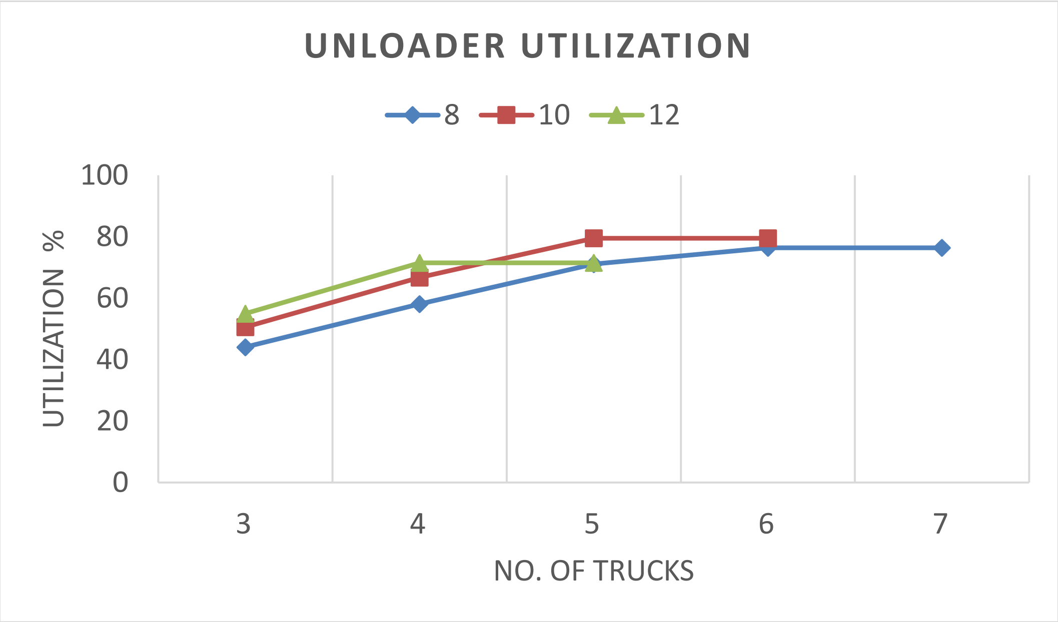


Figure 6-26: Percentage utilization of loader resource for different scenarios

6.3.2 Simulation Model Results of Construction Energy

The simulation model for the construction energy is already discussed in section 5.3. The coefficients C4, C5 and C6 were calculated to reflect the energy consumption per unit time for plant, pump and truck respectively. Table 6-33 shows the data for the placement of concrete using base scenario. The second column shows the duration of the activities and the fourth column gives the coefficient for resources. The total time taken to pour 660 cubic meters of concrete is found to be 1680 minutes, while the energy consumption is 116165 MJ. To optimize the duration and energy consumption, three variants in the capacity of the trucks (8 m^3, 10 m^3 and 12 m^3) with different number of trucks are experimented. The results are tabulated in Table 6-34.

Table 6-33: Base Scenario considerations for Placement of Concrete

Activity	**Duration (mins)**	**Resource/Entity**	**Resource Energy (MJ/min) /Quantity**
Load	22	Plant	18.09 MJ/min
Travel	60	Trucks (10ton)	3.6295 MJ/min
Pour	8	Pump	5.8212 MJ/min
Travel Back	30	Trucks (10ton)	4 nos.

Table 6-34: Duration and energy consumption with three variants in truck capacity and number

Capacity/Nos	**3 trucks**	**4 trucks**	**5 trucks**	**6 trucks**	**7 trucks**	**8 trucks**
8 m^3	--	--	1955 min 137292 MJ	1650 min 136960 MJ	1507 min 137600 MJ	1507 min 137600 MJ

10 m^3	--	--	1680 min 116165 MJ	1550 min 116165 MJ	1550 min 116165 MJ	1550 min 116165 MJ
12 m^3	2375 min 100133 MJ	1800 min 100585 MJ	1475 min 101038 MJ	1475 min 101038 MJ	1475 min 101038 MJ	--

To further analyse these results, similar strategies are adopted as were adopted in the case of transportation model. Figure 6-27 below shows a comparison of pouring time for different capacities of ready-mix trucks of concrete with a varying number of trucks. It can be observed that the green line depicting the 12 m^3 truck achieves the lowest point at 5 number of trucks which is actually the minimum (optimum) time of 1475 minutes. The 10m^3 and 8 m^3 variants achieved their minimum of 1550 min and 1507 min with 6 and 7 number of trucks respectively. Therefore, the 12 m^3 variant remains the most suitable choice as far as optimum time is concerned.

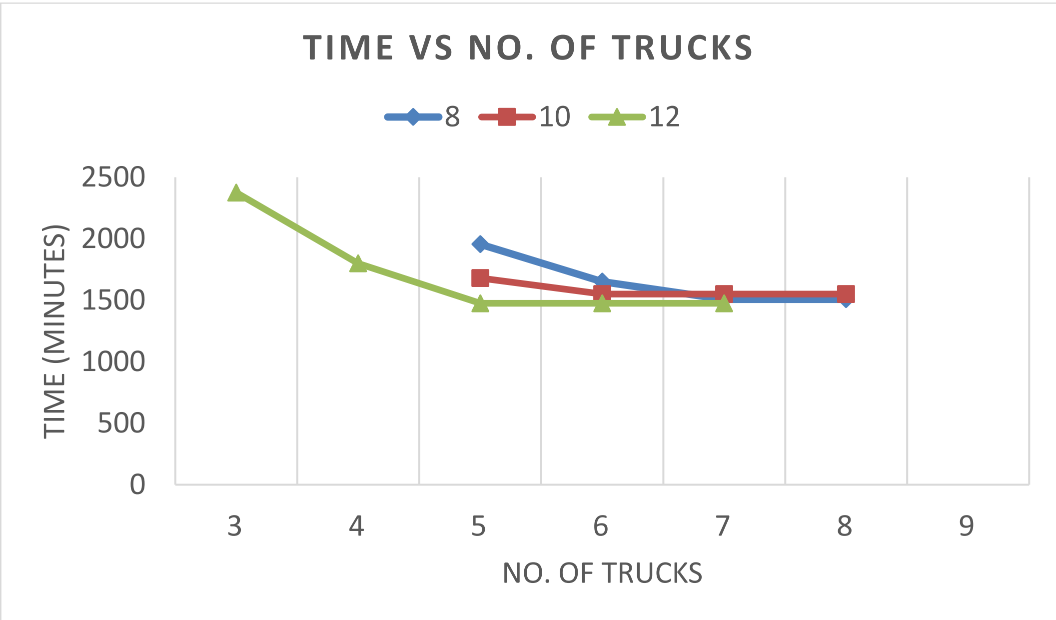


Figure 6-27: Impact of different truck sizes and number of trucks on construction time

As far as the energy is concerned, Figure 6-28 shows a comparison of construction energy for different truck sizes as well as a varying number of trucks. It can be observed that there is little or no impact on energy by varying the number of trucks. The 12 m^3 trucks maintains the lowest energy of 100133 Mega-Joule at 3 number of trucks. However, considering both energy and a 12 m^3 with five number of trucks gives the best choice for this model with a time of 1475 minutes and energy consumption of 101038 Mega-Joule.

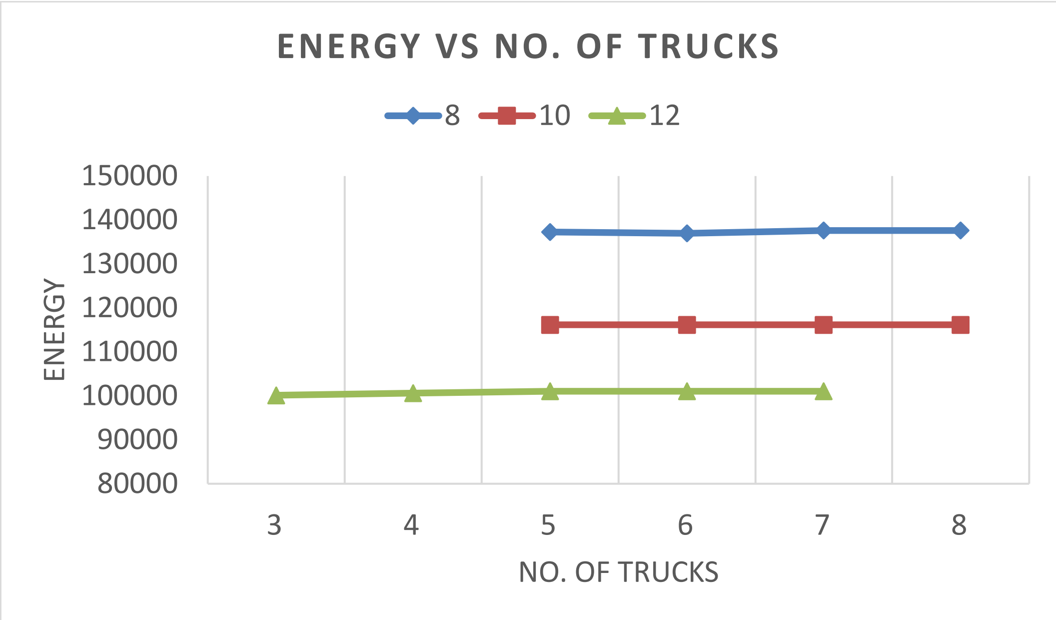


Figure 6-28: Impact of different truck sizes and number of trucks on construction energy

Now, to investigate these results as to why such results were obtained, the auxiliary results of the model must be taken into consideration. The first thing to note is that the 12 m^3 truck achieved its optimum time with only 5 trucks while the other variants achieved their minimum with higher number of trucks. By observing the graphs in Figure 6-29, it can be seen that the graph of 12 m^3 variant for 5 number of trucks smoothened out while the graph of other variants for same number of trucks smoothened out later with respect to number of trucks. This lack of variation depicts constant value which indicates achievement of maximum production rate. Similar findings can be observed in Figure 6-30, that the point at which the production rate value becomes constant with respect to number of trucks, it is the same configuration where production rate chart smooths out.

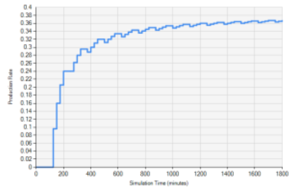
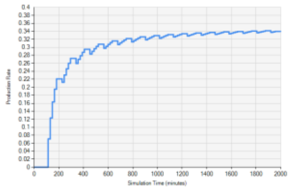
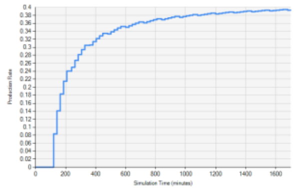
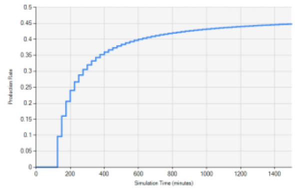
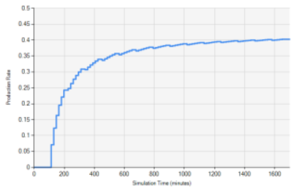
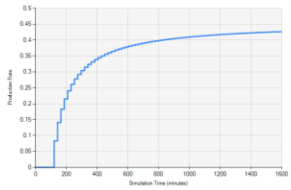
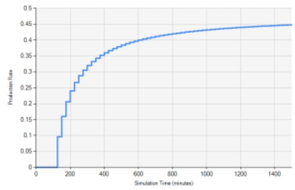
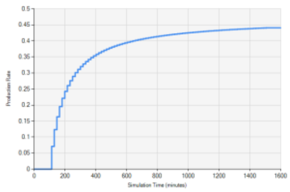
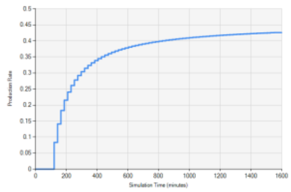
No.	Truck size 8	Truck size 10	Truck size 12
4			
5			
6			
7			

Figure 6-29: Production rates for different truck sized and number of trucks

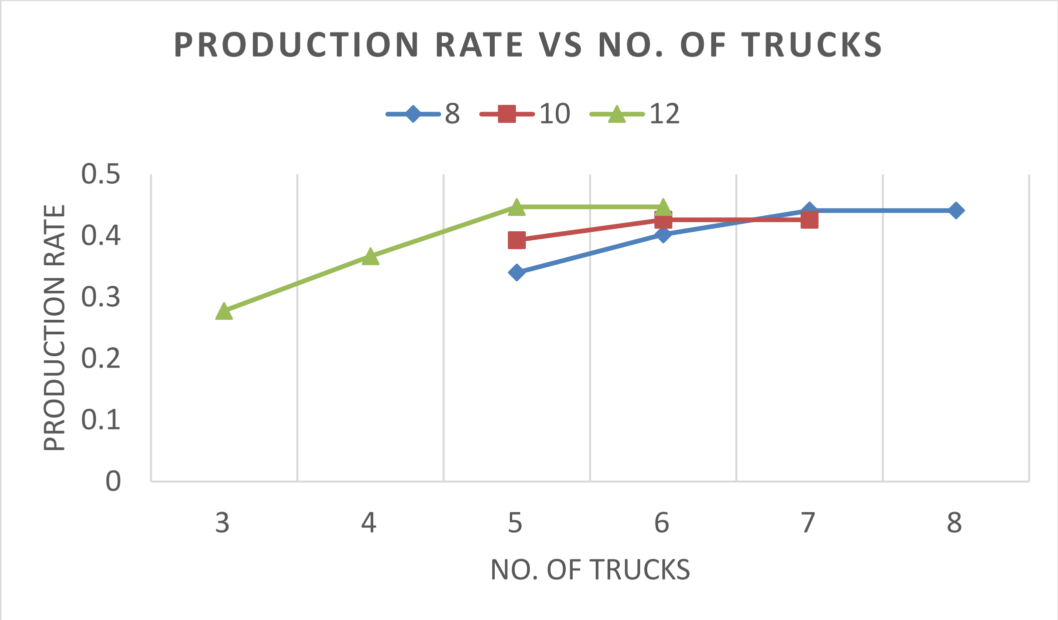


Figure 6-30: Comparison of production rates of different truck sizes and number of trucks

To further investigate as to which resource actually acts as a bottle neck for improving the production rate despite increasing the number of trucks, Figure 6-31 show the utilization graphs of the plant resource. It can be observed that the utilization for the plant resource reaches 100% with only five trucks of the 12 m^3 variant. So, it can be said that one of the constraints in the productivity is the capacity of the loader. Figure 6-32 illustrates a comparison of the utilization of the plant resource with respect to all variants and number of trucks. However, the other resource i.e. the pump, is also looked in for such issues.

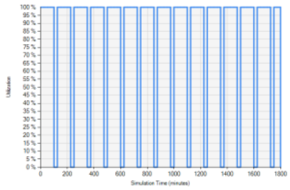
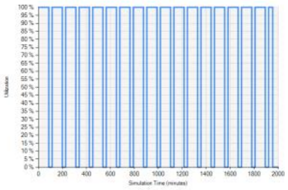
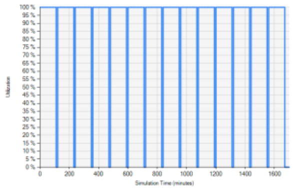
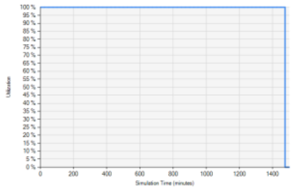
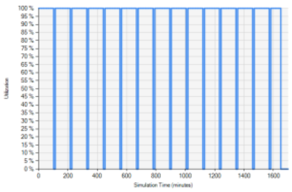
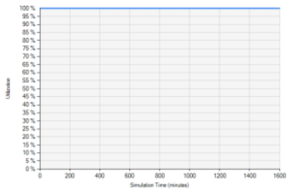
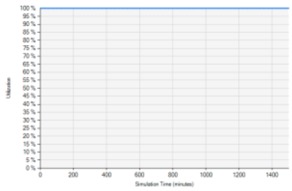
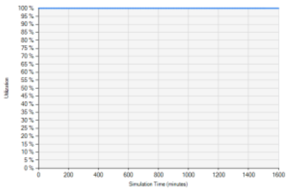
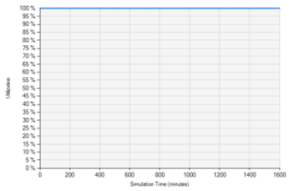
No.	Truck size 8	Truck size 10	Truck size 12
4			
5			
6			
7			

Figure 6-31: Percentage utilization of plant resource for different scenarios

Figure 6-33 presents the utilization of the pump resource, none of the variants are found to achieve 100% utilization of the pump resource. Figure 6-34 illustrates a comparison of the utilization of the pump resource with respect to all variants and number of trucks. The graph is found similar to the loader resource utilization but with around 40% of maximum utilization. These results further strengthen the claim that the major bottle neck appears to be the plant resource.

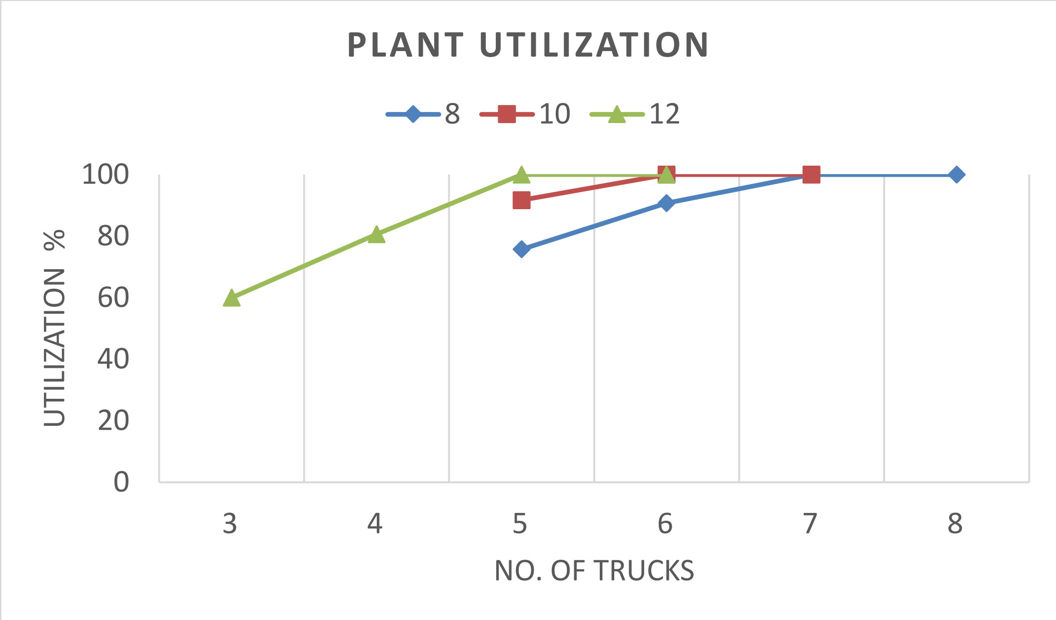


Figure 6-32: Utilization graphs of plant resource

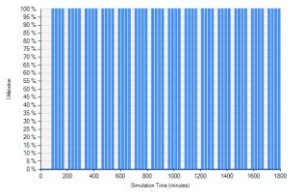
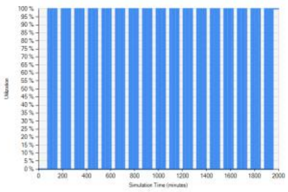
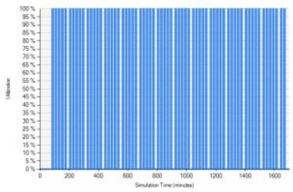
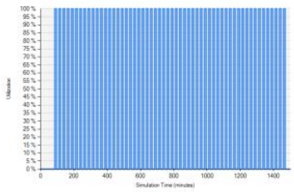
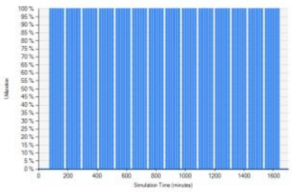
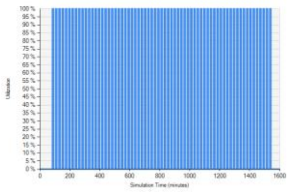
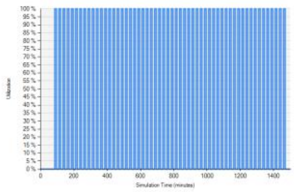
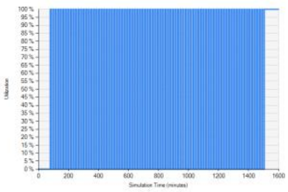
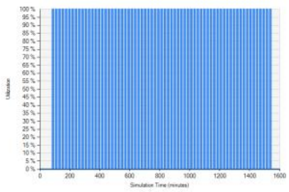
No.	Truck size 8	Truck size 10	Truck size 12
4			
5			
6			
7			

Figure 6-33: Utilization graphs of pump resource

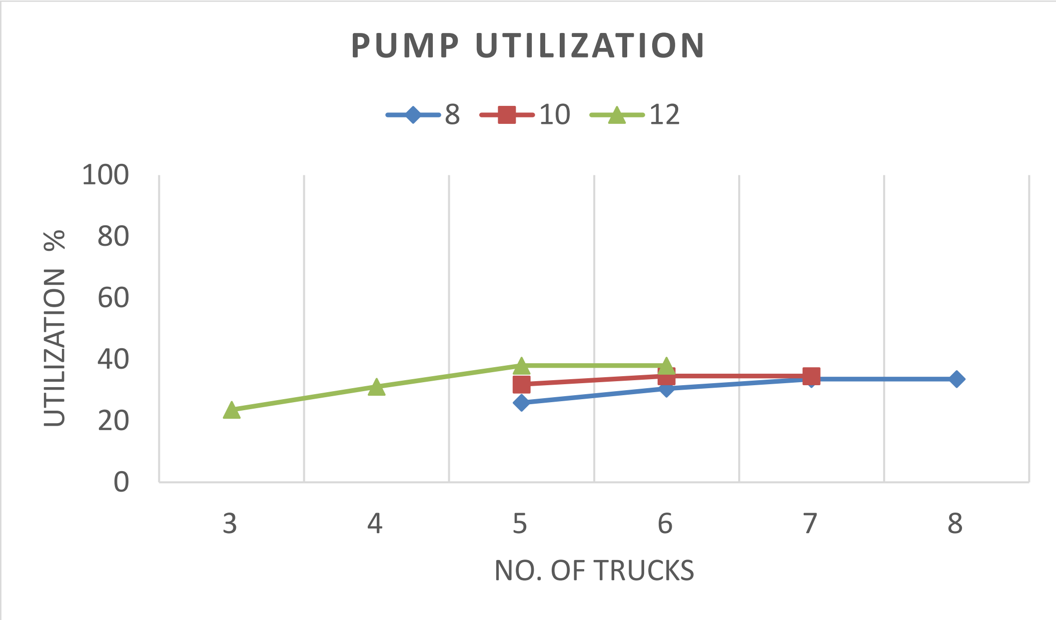


Figure 6-34: Percentage utilization of pump resource for different scenarios

6.4 Enhancing interoperability through IFC

6.4.1 Performance of IFC data exchange

As described in the section 3.5.1, a data exchange experiment is conducted between two BIM authoring tools; Revit and Tekla. Figure 6-35 shows a one room structure of 50 sq. feet. The structure contains four walls, two slabs, four columns, two windows and a door. This model is then exported in IFC format using the Revit Alternate User Interface for IFC for IFC 2X3 coordination view 2.0. Table 6-35 below shows the summary information of the IFC file extracted using IFC File Analyzer. Originally 6 building elements were modelled however IFC export added two more objects namely IfcOpeningElement for windows/doors and IfcCovering for ceiling.

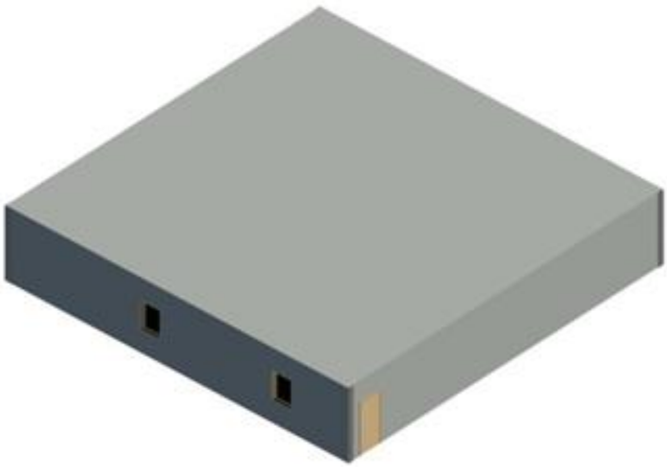


Figure 6-35: A single story model in Revit

Table 6-35: IFC export summary information for Model 1-Revit

Parameter	Value
Revit Model Size	2.56 MB
Physical IFC File Size	37 KB
Number of Entities	588
Object Types Original (Revit)	6
Object Types IFC	8
IFC Export Method Used	20140606_1530(x64) - Exporter 15.4.0.0 - Alternate UI 15.4.0.0
Schema Name	IFC2X3

The IFC file was then imported back to Tekla as a reference model, and then converted to native Tekla model. The Figure 6-36 below shows the steps of conversion. To understand which changes have been to the model, the model was again exported through Tekla as IFC file. Tekla generated an IFC conversion log report to identify the converted elements and the status of conversation, part of that report is shown in Figure 6-37. The Class column represents the adequacy of the information available to convert into native objects. The class value ranges from 990 to 996. 990 depicting that there is enough information in IFC model to convert the objects successfully and 996 showing that there is limited information to convert objects and is actually converted from B-rep (Boundary Representation) object type.

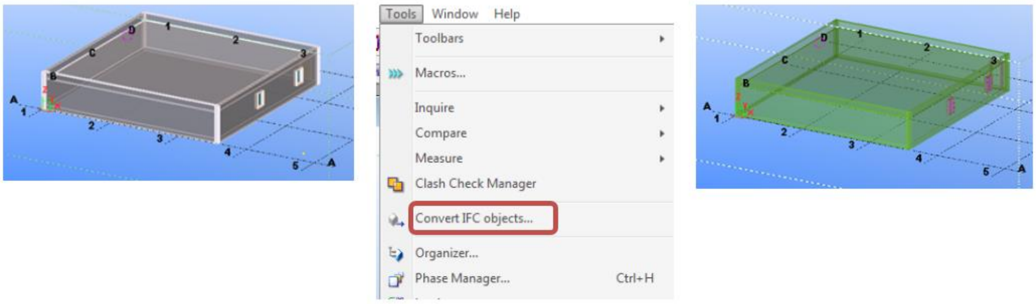


Figure 6-36: Conversion to Tekla native objects

Profile	Initial profile	Class
304.8*15037	GENERIC-12"	990
787.4*12.7	36"X48"	990
787.4*12.7	36"X48"	990
914.4*50.8		990
3048*203.2	GENERIC-8"	990
3048*203.2	GENERIC-8"	990
3048*203.2	GENERIC-8"	990
3048*203.2	GENERIC-8"	990
15036.8*15036.8	2'X4'ACTSYSTEM	990
HD260*73	W10X49	990
HD260*73	W10X49	990
HD260*73	W10X49	990
HD260*73	W10X49	990
36" X 48"		992
36" X 48"-1		992
36" X 48"-2		992
36" X 48"-3		992
36" X 48"-4		992
36" X 48"-5		992
36" X 84"		992
36"X84"	36"X84"	992

Figure 6-37: Part of the Tekla IFC object conversion report

Now the same treatment is performed on the IFC exported from Tekla using IFC analyser, a summary of the result is shown in Table 6-36. The IFC file exported from Revit had 8 object types which now reduced to 4; as the frame/panels for doors/ windows were converted in Beams and the ifcOpening was discarded.

Table 6-36: IFC export summary information for Model 1-Tekla.

Parameter	Value
Tekla Model Size	796 KB
Physical IFC File Size	70 KB
Number of Entities	988
Object Types Modeled (exported from Revit)	8
Object Types IFC	4
IFC Export Method Used	Tekla Structures Educational 20.1 Service Release 2, IFC Export Version: 4.0.0.0 Jan 14 2015
Schema Name	IFC2X3

An alternate method was also employed to import the IFC file through Tekla BIMsight but there was no difference found in any of the procedure followed for this case. Finally, both the IFC files were loaded in SMC for comparison. SMC has a very powerful rule template named Model Comparison (Rule Template # SOL/206) that is able to compare two versions of an IFC model for differences. The final report for the model comparison is summarized in the following Table 6-37.

Table 6-37: Model Comparison Report.

Location	Component	Type	Δ Count	Δ Length	Δ Area
Floor 0	Beam	36" X 48"		0.15	
		36" X 48"-1		0.46	
		36" X 48"-2		0.21	
		36" X 48"-3		0.15	
		36" X 48"-4		0.46	
		36" X 48"-5		0.21	
		36" X 84"		0.08	
		36"X84"		0.08	
	Column	15036.8*15036.8		0.19	
		787.4*12.7		7.17	
		914.4*50.8		7.00	
		HD260*73		40.00	
	Door	36" x 84"	-1		
	Slab	304.8*15037			2433.83
		Generic - 12"			-2433.83
	Suspended	2' x 4' ACT System			-2433.83
	Wall	3048*203.2			1959.14
		Generic - 8"			-1959.14
	Window	36" x 48"	-2		
Level 2	Column	W10X49		-40.00	

Similarly, a two-storey building was modelled on Revit and then exported as IFC. The model is shown in Figure 6-38a, it has Slabs, Walls, Stairs, Railings, Furniture, Openings and Doors. A section detail of the model is shown in Figure 6-38b. This Revit model is exported in IFC format using the Revit Alternate User Interphase for IFC for IFC 2X3 coordination view 2.0. Table 6-38 below shows the summary information of the IFC file extracted using IFC File Analyzer. Originally 7 building elements were modelled however IFC export added two more objects namely IfcCovering for ceiling and IfcMember of stair case stringer.

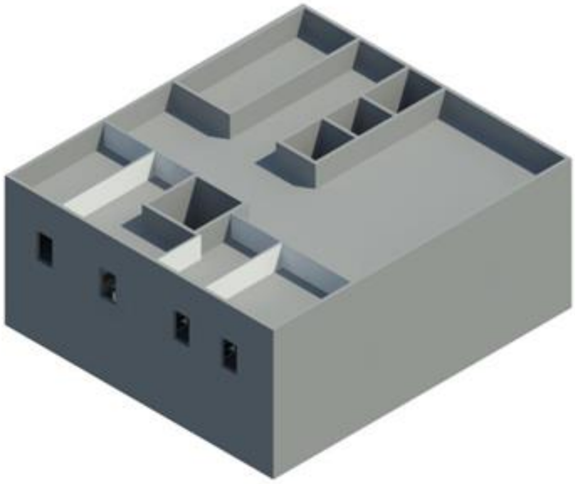


Figure 6-38a: Two Story Revit Model

Figure 6-38b: Section of Two-Story Revit Model

\

Table 6-38: IFC export summary information for Model 2-Revit

Parameter	**Value**

Revit Model Size	5.57 MB
Physical IFC File Size	734 KB
Number of Entities	12539
Object Types Modelled	7
Object Types IFC	9
IFC Export Method Used	20130308_1515 - Exporter 2014.0.2013.0308 - Alternate UI 2.13.0.1
Schema Name	IFC2X3

This IFC file was imported in TEKLA as a reference model and then the reference model was converted into TEKLA native objects. IFC Log report was generated to identify the converted elements and the status of conversation. The IFC Log file is shown in Figure 6-39.

Profile	Initial profile	Class
12.5*1000		990
12.5*1000		990
12.5*1000		990
12.5*1000		990
12.5*1000		990
12.5*1000		990
50*571.76	???-50MM??	990
150*2180	??-150MM	990
750*51		990
750*51		990
750*51		990
750*51		990
1000*250		990
1000*250		990
1000*250		990
1000*250	50MM??13MM??	990
3850*124	??-???100	990
4000*124	??-???100	990
4000*124	??-???100	990
4000*124	??-???100	990
4000*124	??-???100	990
8000*124	??-???100	990
???-50MM??-1	???-50MM??	992
???-50MM??-2	???-50MM??	992
???-50MM??-3	???-50MM??	992
???-50MM??-4	???-50MM??	992
IFC BREP	???	996
IFC BREP-1	1200X2100MM	996
IFC BREP-2	???	996

Figure 6-39: Part of the Tekla IFC object conversion report

As can be seen in the Figure 6-39, three types of classes are displayed; 990 type which shows these objects have sufficient information, 992 type which shows that parametric information may be missing from the elements and the last 996 type showing there is minimum information about the object and it is converted from Boundary representation type object (BREP). Both these models were loaded into Solibiri Model Checker and after using model comparison ruleset the results summarized in Table 6-39 were obtained.

Table 6-39: Model Comparison Report.

Location	Compone	Type	Δ Count	Δ Length	Δ Area
Floor 0	Beam	750 X 2000 MM		0.33	
		750X2000MM		0.33	
		???-50MM??-1		0.16	
		???-50MM??-2		0.16	
		???-50MM??-3		0.16	
		???-50MM??-4		0.16	
	Column	1000*250		3.23	
		12.5*1000		13.01	
		50*571.76		1.31	
		750*51		26.25	
		???-50MM??		1.31	
	Object	IFC BREP	1		
		IFC BREP-1	1		
		IFC BREP-10	1		
		IFC BREP-2	1		
		IFC BREP-3	1		
		IFC BREP-4	1		
		IFC BREP-5	1		
		IFC BREP-6	1		
		IFC BREP-7	1		
		IFC BREP-8	1		
		IFC BREP-9	1		
	Railing	900mm 圆管	-2		
	Slab	150*16538			1701.24
		150*16562			2979.77
		150*2180			77.39
		150*5888			856.91
		50*1020			28.95
		常规 - 150mm			-2979.77
		非整体平台			-28.95
	Stair	190mm 最大踢面 250mm	-1		
		50mm 踏板 13mm 踢面	-2		
	Suspende	无装饰			-2578.82
	Wall	3850*124			312.37
		4000*124			860.57
		8000*124			1145.50
		8000*200			4058.32
		8150*200			5393.47
		8150*300			728.07
		内部 - 砌块墙 100			-1703.28
		常规 - 200mm			-9451.79
		常规 - 300mm			-728.07
标高 2	Door	1200 x 2100mm	-1		
		750 x 2000 mm	-4		
	Furniture	1830 x 915 mm	-7		
		玻璃柜	-5		
		转椅 9	-7		
	Slab	常规 - 150mm			-2635.54
	Suspende	无装饰			-2928.32
	Wall	内部 - 砌块墙 100			-615.16

An alternate method was also employed to import the IFC file through Tekla BIMsight and then storing in *.tbm format. The model was imported in Tekla as Tekla BimSight project and converted into native tekla objects. IFC Log report was generated to identify the converted elements and the status of conversation. The IFC Log file showed improvement as class 996 was eliminated and previously 28 objects were in class 992 and 996 and now only 17 objects are in class 992. The models were compared in the Solibiri Model Checker and the analysis showed similar results but without the BREP objects as these were understood by the tekla in this case.

An alternate method was also employed to import the IFC file through Tekla BIMsight and then storing in *.tbm format. The model was imported in Tekla as Tekla BimSight project and converted into native tekla objects. IFC Log report was generated to identify the converted elements and the status of conversation. The IFC Log file showed improvement as class 996 was eliminated and previously 28 objects were in class 992 and 996 and now only 17 objects are in class 992. The models were compared in the Solibiri Model Checker and the analysis showed similar results but without the BREP objects as these were understood by the tekla in this case.

6.4.2 IFC file with embodied energy embedment

A theoretical extension of IFC to embed energy data is presented in section 4.4, and the it implementation by modifying the xBIM toolkit is presented in section 5.4. This section addresses the form and utility of the resulting IFC data. As far as the form is concerned, Figure 6-40 below shows parts of the IFC file created after adding the embodied energy values. Since the resulting file is quite large only representative sections are presented in the figure. The first section represents the material energy information for a single wall and only concrete. The *IfcRelAssignsToControl* binds the *IfcEnergyItem* to the respective *IfcWall* element. Similarly, the next line binds the *IfcEnergyValue* with the *IfcEnergyItem*. This step wise integration continues till the the value with unit is bound to the *IfcWall*. Finally, *IfcRelNests* binds all the contribution to a single energy item which acts as the total embodied energy.

```
/* Energy for one Wall*/

/*Material Energy (Only Concrete)*/
#83973=IFCRELASSIGNSTOCONTROL('3wOQ7uTSD9pfsZ6abgYvFw',#83874,$,$,(#1229),$,#83974);
#83974=IFCENERGYITEM('2qv4WxeVr15A_i8zkD88ox',#83874,'Material',$,$,$,.USERDEFINED.,(#83975),$);
#83975=IFCENERGYVALUE('Concrete','Cast-in-Place gray',IFCRATIOMEASURE(1.),#83976,$,$,$,$,.ADD.,$);
#83976=IFCMEASUREWITHUNIT(IFCENERGYMEASURE(15112.65),#83977);
#83977=IFCSIUNIT(*,.ENERGYUNIT.,$,.JOULE.);
.
.
.

/*Transportation Energy (One Wall, Only Concrete)*/
#103978=IFCRELASSIGNSTOCONTROL('2Yhc8jv_P52e4vix14$J73',#83874,$,$,(#1229),$,#103979);
#103979=IFCENERGYITEM('1OrxtqC35B4OSAQyemsa_1',#83874,'Transportation',$,$,$,.USERDEFINED.,(#103980),$);
#103980=IFCENERGYVALUE('Concrete','Wall.1',IFCRATIOMEASURE(1.),#103981,$,$,$,$,.ADD.,$);
#103981=IFCMEASUREWITHUNIT(IFCENERGYMEASURE(342.19),#103982);
#103982=IFCSIUNIT(*,.ENERGYUNIT.,$,.JOULE.);
.
.
.

/*Construction Energy (One Wall, Only ConcreteMixer*/
#123983=IFCRELASSIGNSTOCONTROL('2F5i7aIiH5qhMktIqNwG5j',#83874,$,$,(#1229),$,#123984);
#123984=IFCENERGYITEM('0Sv4LLFSD32RVKScla9Ha9',#83874,'Construction',$,$,$,.USERDEFINED.,(#123985),$);
#123985=IFCENERGYVALUE('ConcreteMixer','',IFCRATIOMEASURE(1.),#123986,$,$,$,$,.ADD.,$);
#123986=IFCMEASUREWITHUNIT(IFCENERGYMEASURE(9.85),#123987);
#123987=IFCSIUNIT(*,.ENERGYUNIT.,$,.JOULE.);
.
.
.

/*Total Energy (One Wall)*/
#163988=IFCRELASSIGNSTOCONTROL('2DT6NOAoPC5eY$mG0$B5mE',#83874,'Total',$,(#1229),$,#163989);
#163989=IFCENERGYITEM('1_iKSoN2bCpQI0DpAv$MUm',#83874,'Total EE',$,$,$,.USERDEFINED.,(#163990),$);
#163990=IFCENERGYVALUE('Wall.1','',IFCRATIOMEASURE(1.),$,$,$,$,$,.ADD.,$);
#163991=IFCMEASUREWITHUNIT(IFCENERGYMEASURE(28695.03),#163992);
#163992=IFCSIUNIT(*,.ENERGYUNIT.,$,.JOULE.);
.
.
.

/* Total Energy = Material Energy + Construction Energy + Transportation Energy*/
#18993=IFCRELNESTS('2M9WD9Ps91_vTL$BgGjonR',#83874,$,$,#163989,(#83974.................#103979.................#123984..........));
```

Figure 6-40: Resulting IFC file after adding the embodied energy information

To understand the utility of the embedment of energy values in the IFC data structure, first the efficacy of the IFC data itself is needed to be appreciated. The IFC data format provides an interoperable medium through which different software packages developed by various vendors can create and access data without any proprietary restrictions. Therefore, embedding energy data in the IFC file makes it highly interoperable and available for further analysis as well as adequate presentation.

6.5 Summary

This chapter discusses the implementation of the framework on a case study. Firstly, the data collection phase is discussed then the results of assessment of embodied energy through the prototype is presented. Subsequently, validation of the results is conducted in the form of detailed manual calculations. Then some additional assessment outside the scope of the prototype but through the available data set is given. Later, the results for simulation of transportation and construction energy are shared. Finally, the working of the export of embodied energy data in the IFC format is shown.

CHAPTER 7: DISCUSSION AND CONCLUSION

The research presented in this thesis aimed to propose a practical and workable solution for considering sustainability information as the critical factor for evaluating building alternatives. Traditionally, cost and time had been the critical factors to assess the success of a construction project, but recently, the performance of construction projects is being also tied with sustainability. Nevertheless, the methods for estimation of sustainability quotients such as embodied energy are far too complex to be adopted in routine construction projects. Not only these methods are rigorous in their approach but also need information from multiple sources. BIM provides a platform to integrate all such information in a single environment and enable further development to analyse the building data for various applications.

This study therefore, (i) Investigated the total energy consumption in delivering building as a product using BIM, (ii) Proposed a method to automatically calculate the energy consumption including embodied energy in material, energy consumption in transportation and energy consumption in construction, (iii) Provided suggestions for energy saving based on transportation and construction process simulation analysis and (iv) Enhanced the interoperability of sustainability data by extending the current IFC model to incorporate embodied energy information.

7.1 Discussion on investigation of the total energy consumption in delivering building as a product using BIM

A case project is investigated for the total energy consumption which helps the decision makers to identify those areas having a major impact on the embodied energy content of the project. For instance, Figure 6-15 indicates that almost all of the embodied energy is contributed by the material energy. This is understandable because if a building is considered as a product, then transportation and construction are merely processes to build this final product. However, any material is itself a product of different processes and therefore is entitled to higher energy demands. Similarly, Figure 6-16 identifies that foundations constitute more than half of the material embodied energy. And by looking at Table 6-8, it becomes evident that in the context of the foundation, the contribution from the concrete is on the much higher side. In contrast to this finding, the next contender in Figure 6-16 are floors, and by looking at Table 6-8, it seems that in the context of floors the contribution from steel is quite close to that of concrete.

Consequently, Figure 6-17 illustrates that concrete dominates the transportation energy domain, and the cause may be found in Table 6-9. Apart from the obvious greater quantity of the concrete that needs to be transported, the fuel consumption coefficient also plays an important part. Subsequently, Figure 6-18 demonstrates the domination of the concrete pump in the domain of construction energy, and by looking at Table 6-10 it appears that the use of a concrete pump for foundation works contributes about 50%. Other noteworthy contributors in Figure 6-18 are the excavator and bulldozer, by looking at Table 6-16 it seems that both equipment only contributed towards the foundation. The above discussion reveals that foundation works are dominant in all three contributors of the embodied energy. Thus, if significant efforts are made to reduce the impact of foundation works, the overall contribution of embodied energy can be lowered.

The major advantage of the tool is that it works within the BIM environment and all the information entered remains a part of the database and can be retrieved anytime within the BIM environment. Thus, any optimization approach or algorithm can be used in the current context to limit the embodied energy content.

7.2 Discussion on method to automatically calculate the embodied energy consumption using BIM

This research investigated BIM as a platform to integrate information about energy consumption in delivering building as a product. A full understanding of energy consumption has been gained through the literature review and the calculation of energy consumption through a case study. External database is created to save data and integrated with BIM model for an integrated analysis through a developed user interface. A methodology is proposed in this study to cover the procedure of data collection, data mapping and embodied energy consumption analysis. Material quantity retrieved from BIM model provides the fundamental data for all analysis, based on which, energy calculation of transportation and construction can be carried out.

The data extracted from the project documents provide extra information for individual project to make the energy consumption analysis more complete and closer to real situations. After the realization of this data in the BIM model, the material, transportation and construction energies are estimated using a prototype tool. The novelty of the current approach rests on the pillars of integration and automation. The previous studies either disregard the importance of integration; none of the studies comprehensively estimated the embodied energy within the BIM environment, or rely on human interventions to execute the embodied energy estimation process causing a potential of errors and lack of automation.

7.3 Discussion on energy saving based on transportation and construction simulation models.

In this study, two representative simulation models are developed in Simphony package to simulate and optimize the transportation and construction energy. Although appreciable amount of literature is available for simulation and optimization of durations, but limited research is done on the simulation of embodied energy. So, the novelty aspect here is to derive the mathematical equations to obtain the embodied energy coefficients that can be used in the simulation model package. The first model simulated the transportation of steel bars, comprising of circulating entities "Trucks" and two resources loader and unloader. Given the constraints of the model, a 12-ton truck variant with 4 number of trucks was found to be most suitable in terms of both energy and duration.

Similarly, the second representative model was developed for pouring of concrete. Ready-mix concrete trucks were set as the circulating entities with two resources of pump and plant. A 12 m^3 truck variant with 5 number of trucks was found to be most suitable in terms of both energy and duration. A substantial difference of energies was found among the energies of different scenarios which indicates the effectiveness of the technique.

7.4 Discussion on embedment of embodied energy in IFC model

The sustainability information is not adequately mapped on standard construction information models such as IFC. Therefore, this study proposed an extension for the IFC standard to embed the embodied energy information in the IFC model. This was achieved by proposing a new resource "*IfcEnergyResource*" with a new data type "*IfcEnergyValue*". To assign this data type to different resources, two new shared management elements, "*IfcEnergyItem*" and "*IFCEnergySchedule*", are proposed. *IfcEnergyItem* hold the embodied energy impact of a material, while

IfcEnergySchedule brings together different instance of *IfcEnergyItem*. The schema for *IfcConstuctionMgmtDomain* was also modified to embed the embodied energy information at the domain level so that it can be assigned to various resources abstracted by *IfcConstructionResource.*

This extension will aid in bringing back the sustainability information in the BIM model from the LCA tools and restores the authenticity of the central BIM model as a knowledge base in the context of sustainability as well. Apart from that this theoretical extension, this study also implemented the extension by modifying an open source tool (xBIM) to create the above mentioned entities and then merged this extension in the prototype tool to practically embed embodied energy information in the IFC model.

7.5 Conclusion

In conclusion, solid contributions to the body of knowledge are highlighted:

1. A novel framework is proposed to assess embodied energy for the building construction project using BIM and implemented by designing and developing a BIM based tool.

2. A novel method for embodied energy simulation and optimization with respect to transportation and construction is proposed and implemented.

3. An extension to the underlying BIM data structure (IFC) is proposed and implemented to embed embodied energy content with respect to material transportation and construction energy.

7.6 Limitation and Future Work

The current approach lacks in handling multiple design scenarios as well as in providing meaningful suggestions to improve the sustainability performance of the building design and construction. In addition, the data source for the embodied energy coefficients needs to be revisited as these coefficients vary for different regions and should somehow be linked with national statistics. Apart from that, the simulation models used in this study are simple and of course far from professional effectiveness. These models are developed just to demonstrate the applicability of the framework and much more complex, comprehensive as well as practical models previously exist for simulating cost and time. In future, if these limitations are capitalized and a decision framework system is designed to take in all the available data and automatically generate what-if scenarios as well as evaluate their sustainability performance, then such an efficient system can aid in creating a better future.

REFERENCES

[1] IEA, Key World Energy Statistics, International Energy Agency, Paris, France, 2013.

[2] B. Metz, O.R. Davidson, P.R. Bosch, R. Dave, L.A. Meyer. Contribution of working group III to the fourth assessment report of the intergovernmental panel on climate change, 2007.

[3] R. Crawford, G. Treloar. An assessment of the energy and water embodied in commercial building construction, Proceedings of Australian Life Cycle Assessment Conference, Sydney, Australia, 2005, pp. 1–10.

[4] M.Y. Han, G.Q. Chen, L. Shao, J.S. Li, A. Alsaedi, B. Ahmad. Embodied energy consumption of building construction engineering: case study in E-town, Beijing, Energy Build. 64 (2013) pp. 62–72.

[5] G.Aranda-Mena, J. Crawford, A. Chevez, T. Froese. Building information modelling demystified: does it make business sense to adopt BIM? Int. J. Manag. Projects Bus. 2 (3) (2009) pp. 419–434.

[6] R. Takim, M. Harris, A.H. Nawawi. Building Information Modeling (BIM): a new paradigm for quality of life within architectural, engineering and construction (AEC) industry, Procedia – Social Behav. Sci. 101 (2013) pp. 23–32.

[7] F.H. Abanda, A.H. Oti, J.H.M. Tah. Integrating BIM and new rules of measurement for embodied energy and CO 2 assessment, J. Build. Eng. 12 (2017) pp. 288–305.

[8] T. Liebich. IFC 2× Edition 3 Model Implementation Guide (Version 2.0) 2010

[9] Z. Alwan, P. Jones. The importance of embodied energy in carbon footprint assessment, Struct. Surv. 32 (1) (2014) pp. 49–60.

[10] X. Wei, L.C. Guo. Analysis of building construction energy consumption, Constr. Technol. 43 (16) (2014) 74–77 doi: 1002- 8498(2014)16- 0074- 04.

[11] J. Goggins , T. Keane , A. Kelly. The assessment of embodied energy in typical reinforced concrete building structures in Ireland, Energy Build. 42 (5) (2010) pp. 735–744.

[12] C.K. Chau , T.M. Leung , W.Y. Ng. A review on life cycle assessment, life cycle energy assessment and life cycle carbon emissions assessment on buildings, Appl. Energy 143 (2015) pp. 395–413.

[13] G. Hammond, C. Jones. Inventory of Carbon and Energy (ICE), Version 1.5a Beta, Carbon Vision Buildings Program, University of Bath, UK, 2006

[14] C.K. Chau, J.M. Xu, T.M. Leung, W.Y. Ng. Evaluation of the impacts of end-of life management strategies for deconstruction of a high-rise concrete framed office building, Appl. Energy 185 (2) (2017) pp. 1595–1603 (ISSN 0306-2619).

[15] T. Bassett, D. Waldron, H. Iorwerth, S.C. Lannon, P.J. Jones. Embodied energy at an urban scale: a paradigm shift in calculations, PLEA2013 – 29th Conference, Sustainable Architecture for a Renewable Future, Munich, Germany, 10–12 September 2013, (2013).

[16] D. Yu, H. Tan, Y. Ruan. A future bamboo-structure residential building prototype in China: life cycle assessment of energy use and carbon emission, Energy Build. 43 (10) (2011) pp. 2638–2646.

[17] G. Capper, J. Matthews, S. Lockley, Incorporating embodied energy in the BIM process, CIBSE ASHRAE Technical Symposium, 18–19 April 2012, Imperial College, London, UK, 2012.

[18] J. Monahan, J.C. Powell. An embodied carbon and energy analysis of modern methods of construction in housing: a case study using a lifecycle assessment framework, Energy Build. 43 (1) (2011) pp. 179–188.

[19] A.M. Moncaster, K.E. Symons. A method and tool for ‘cradle to grave’ embodied carbon and energy impacts of UK buildings in compliance with the new TC350 standards, Energy Build. 66 (2013) pp. 514–523.

[20] A. Artenian, F. Sadeghpour, J. Teizer. A GIS framework for reducing GHG emissions in concrete transportation, Construction Research Congress 2010, American Society of Civil Engineers, 2010, pp. 1557–1566.

[21] Y. Chang, R.J. Ries, Q. Man, Y. Wang. Disaggregated I-O LCA model for building product chain energy quantification: A case from China, Energy and Buildings, Vol. 72, 2014, pp. 212-221, ISSN 0378-7788, https://doi.org/10.1016/j.enbuild.2013.12.026.

[22] J. Nässén, J. Holmberg, A. Wadeskog, M. Nyman. Direct and indirect energy use and carbon emissions in the production phase of buildings: An input–output analysis, Energy, Vol. 32, Issue 9, 2007, pp. 1593-1602, ISSN 0360-5442, https://doi.org/10.1016/j.energy.2007.01.002.

[23] G.J. Treloar. Extracting Embodied Energy Paths from Input-Output Tables: Towards an Input-Output-based Hybrid Energy Analysis Method, Economic Systems Research, Vol. 9, 1997, No. 4, pp. 375- 391

[24] A. Stephan, L. Stephan, Reducing the total life cycle energy demand of recent residential buildings in Lebanon, Energy, Vol. 74,2014, pp. 618-637, ISSN 0360-5442

[25] A. Guggemos, A. Horvath, Decision-support tool for assessing the environmental effects of constructing commercial buildings, J. Archit. Eng. 12 (4) (2006) pp. 187–195.

[26] A. Sharrard, H. Matthews, M. Roth, Environmental implications of construction site energy use and electricity generation, J. Construct. Eng. Manage. 133 (11) (2007) pp. 846–854.

[27] R.H. Crawford, G.J. Treloar, Assessment of embodied energy analysis methods for Australian construction industry, in: The 38th International Conference of Architectural Science Association ANZAScA, Tasmania, 2004.

[28] A. Stephan & R.H. Crawford. A multi-scale life-cycle energy and greenhouse-gas emissions analysis model for residential buildings, Architectural Science Review, 57:1, 2014, pp. 39-48, DOI: 10.1080/00038628.2013.837814

[29] Zhao, C. and Li, L. "Exploring a preferable workflow for conducting LCA in a BIM environment" Master's Thesis 2015:108. Chalmers University of Technology.

[30] NBIMS. "National BIM standard purpose". The US National Institute of Building Sciences Facilities Information Council, BIM Committee, 2010.

[31] S. Habibi, The promise of BIM for improving building performance, Energy and Buildings, Vol. 153, 2017, pp. 525-548, ISSN 0378-7788, https://doi.org/10.1016/j.enbuild.2017.08.009.

[32] C. Zhang, J. Beetz, & M. Waise. Interoperable validation for IFC building models using open standards. Journal of Information Technology in Construction, 2015. ISSN 1874-4753 http://www.itcon.org/2015/2itials

[33] S. Shrivastava, A. Chini, Using building information modeling to assess the ini- tial embodied energy of a building, Int. J. Constr. Manag. 12 (1) (2012) pp. 51–63 .

[34] F.H. Abanda , G.E. Nkeng , J.H.M. Tah , E.N.F. Ohandja , M.B. Manjia , Embodied energy and CO 2 analyses of mud-brick and cement-block houses, AIMS Energy J. 2 (1) (2014) pp. 18–40.

[35] F. Shadram, T.D. Johansson, W. Lu, J. Schade and T. Olofsson. An integrated BIM-based framework for minimizing embodied energy during building design. J. Energy and Buildings Vol. 128, 2016, pp. 592–604.

[36] A. Jrade , R. Abdulla , Integrating building information modelling and life cycle assessment tools to design sustainable buildings, in: International Conference OF CIB W78, 2012, pp. 29 .

[37] Athena Sustainable Material Institute, Athena EcoCalculator, 2012. from http: //www.athenasmi.org/our- software- data/ecocalculator.

[38] F.H. Abanda, B. Kamsu-Foguem, J.H.M. Tah, BIM - new rules of measurement ontology for construction cost estimation, Eng. Sci. Technol. 20 (2) (2017) 443–459.

[39] Tally. A joint development project from KT Innovations, thinkstep, and Autodesk. 2018 KT Innovations. http://choosetally.com/

[40] A. Jrade, F. Jalaei. Integrating building information modelling with sustainability to design building projects at the conceptual stage, Build. Simul. 6 (4) (2013) pp. 429–444.

[41] S. Lee, S. Tae, S. Roh, T. Kim. Green template for life cycle assessment of buildings based on building information modeling: focus on embodied environmental impact, Sustainability 7 (2015) pp. 16498–16512.

[42] S.O. Ajayi, L.O. Oyedele, B. Ceranic , M. Gallanagh , K.O. Kadiri. "Life cycle environmental performance of material specification: a BIM-enhanced comparative assessment, Sustain. Build. Technol. Urban Dev. 6 (2015) pp. 14–24.

[43] C. Peng. Calculation of a building's life cycle carbon emissions based on Ecotect and building information modeling, Clean. Prod. 112 (2014) pp. 453–465 .

[44] Y. Shin, K. Cho, BIM application to select appropriate design alternative with consideration of LCA and LCCA, Math. Probl. Eng. (2015). http://dx.doi.org/10. 1155/2015/281640 . Hindawi Publ. Corp.

[45] S.M. AbouRizk, S.A.Hague and R. Ekyalimpa, Construction Simulation: An Introduction Using Simphony. 2016 University of Alberta. ISBN: 978-1-55195-357-1

[46] L.Y. Shena, W.S. Lua, H. Yaoa, D.H. Wu, A computer-based scoring method for measuring the environmental performance of construction activities, Autom. Constr. 14 (2005) pp. 297–309.

[47] S.S. Sihabuddin, S.T. Ariaratnam, Methodology for estimating emissions in underground utility construction operations, J. Eng. Design Technol. 7 (1) (2009) pp. 37–64.

[48] V. González, T. Echaveguren, Exploring the environmental modeling of road construction operations using discrete-event simulation, Autom. Constr. 24 (2012) pp. 100–110.

[49] A.M. Law, W.D. Kelton, Simulation Modeling and Analysis, 3 rd ed. McGraw-Hill, New York, 2000.

[50] Martínez, J. C., 2010. Role of Simulation in Construction Engineering and Management. Journal of Construction Engineering and Management. 3(7) pp. 1140-1153.

[51] J.C. Martínez, Earthmover — simulation tool for earthwork planning, in: D.J. Medeiros, E.-F. Watson, J.S. Carson, M.S. Manivannan (Eds.), Proceedings of the 30th Winter Simulation Conference, Washington D.C, 1998, pp. 1263–1272.

[52] M. Marzouk, O. Moselhi, Object-oriented simulation model for earthmoving operations, Journal of Construction Engineering and Management 129 (2) (2003) pp. 173–181.

[53] C. Ahn, J.C. Martinez, P.V. Rekapalli, F.A. Peña-Mora, Sustainability analysis of earthmoving operations, in: M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, R.G. Ingalls (Eds.), Proceedings of 41th Winter Simulation Conference, Austin, 2009, pp. 2605–2611.

[54] C. Ahn, W. Pan, S. Lee, F.A. Peña-Mora, Enhanced estimation of air emissions from construction operations based on discrete-event simulation, Proceedings of the International Conference on Computing in Civil and Building Engineering, Nottingham, U.K., 2010.

[55] D.W. Halpin. “CYCLONE: Method for modeling of job site processes.” Journal of the Construction Division. 103(3), 1997, pp. 489–499.

[56] P.G. Ioannou and J.C. Martinez, "Comparison of Construction Alternatives Using Matched Simulation Experiments." Journal of Construction Engineering and Management. 112(3), 1996, pp. 234-241.

[57] J.C. Martínez. "Methodology for Conducting Discrete-Event Simulation Studies in Construction Engineering and Management." Journal of Construction Engineering and Management. 136(1), 2001, pp. 3-16.

[58] S.M. AbouRizk and D. Hajjar. "A framework for applying simulation in the construction industry." Canadian Journal of Civil Engineering. 25(3), 1998, pp. 604–617.

[59] S.M. AbouRizk and Y. Mohamed. "Simphony: An integrated environment for construction simulation." Proc., Winter Simulation Conf., Institute of Electrical and Electronics Engineers, Piscataway, NJ, 2000, pp.1907–1914.

[60] L. Qing, G. Tao and WJ. Ping. 2013. Study on Building Lifecycle Information Management Platform Based on BIM. Research Journal of Applied Sciences, Engineering and Technology 7(1), 2013, pp. 1-8.

[61] M. Laakso, and A. Kiviniemi, "The Ifc Standard - A Review Of History, Development, And Standardization", ITcon., Vol. 17, 2012, pp. 134.

[62] IFC Specification from: http://www.buildingsmart-tech.org/specifications/ifc-overview [Retrieved on Nov 16, 2017]

[63] Chapter 9, Building SMART (n.d). Retrieved on Nov 16, 2017 from: http://buildingsmart.com/organization/

[64] R.S. Nizam and C. Zhang, "Current State of Information Exchange between the two most popular BIM software: Revit and Tekla". Proceedings of the 1st International Conference on Sustainable Buildings and Structures (Suzhou, P.R. China, 29 October - 1 November 2015)

[65] buildingSMART International Ltd. Industry Foundation Classes Release 4 (IFC4) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm [Retrieved on Dec 1, 2018]

[66] buildingSMART International Ltd. IFC Technology. http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-technology [Retrieved on Dec 1, 2018]

[67] G. Lee, J. Won , S. Ham, and Y. Shin, “Metrics for quantifying the similarities and differences between IFC”, J.Comput. Civil. Eng., 2011.

[68] J. Wix, “What is IFC?” retrieved from: www.it.civil.aau.dk/it/education/reports/building_smart/WS3_IDM_WhatIsTheIFCModel.pdf [n.d]

[69] ifcXML Implementation Guide, 2007. Retrieved on Nov 16, 2017 from: http://www.buildingsmart-tech.org/downloads/accompanying-documents/guidelines/ifcXML%20Implementation%20Guide%20v2-0.pdf

[70] A. Motamedi, M.M. Soltani and A. Hammad, “Localization of RFID-equipped assets during the operation phase of facilities”, Adv. Eng. Inform., Vol. 27, no. 4, pp. 566– 579, 2013.

[71] M.M. Soltani, A. Motamedi and A. Hammad, “Enhancing cluster-based RFID tag localization using artificial neural networks and virtual reference tags”, J. Autom. Constr. Vol. 54, pp. 93–105, 2015.

[72] A. Motamedi, M.M. Soltani, S. Setayeshgar, and A. Hammad, “Extending IFC to incorporate information of RFID tags attached to building elements.”. J. Advan. Eng. Inform., Vol. 30, no 1, 2016.

[73] Z. Ma, Z. Wei, and X. Zhng, “Semi-automatic and specification-compliant cost estimation for tendering of building projects based on IFC data of design model,”. J. Automat. Construct., Vol. 30, pp. 126–135, 2013.

[74] IFC4 Add2. “Introduction.” [Online]. Available at: http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/introduction.htm, Accessed on: 31/01/2018.

[75] IFC4 Add2. “Figure 274-Cost Composition” [Online]. Available at: http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifccostitem.htm, Accessed on 31/01/2018.

[76] IFC4 Add2. “Figure 275-Cost Assignment.” [Online]. Available at: http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/ifccostitem.htm, Accessed on 31/01/2018.

[77] J. Hou , Chinese Construction Handbook, 2nd ed., China Machine Press, Beijing, 2009 .

[78] C. Eastman C., J. Lee , Y. Jeong and J. Lee. Automatic rule-based checking of building designs, Journal of Automation in Construction, 18 , 2009, pp. 1011-1033.

[79] IFC4 Add2. Figure 1 — Data schema architecture with conceptual layers [Online]. Available at: http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/link/introduction.htm Accessed on 31/01/2018.

[80] S. Lockley, C. Benghi and M. Černý. Xbim.Essentials: a library for interoperable building information applications. Journal of Open Source Software, 2(20), 2017, pp. 473, https://doi.org/10.21105/joss.00473

[81] Autodesk, Autodesk Developer Network, 2017. Internet http://usa.autodesk. com/adsk/servlet/index?siteID=123112&id=24 84 975.

[82] Microsoft, Windows Forms Overview, 2017. Internet https://docs.microsoft. com/en-us/dotnet/framework/winforms/windows-forms-overview

[83] Baidu, Baidu Maps, 2016. Internet https://map.baidu.com

LIST OF PUBLICATIONS

Journal Papers

1. **Nizam, R.S**., Zhang C. & Tian L. A BIM based tool for assessing embodied energy for buildings. *Journal of Energy and Buildings.* Vol. 170, 2018, pp. 1-14, ISSN 0378-7788. https://doi.org/10.1016/j.enbuild.2018.03.067

2. Zhang C., **Nizam, R.S.**, & Tian L. BIM based investigation of total energy consumption in delivering building products. *Journal of Advanced Engineering Informatics.* Vol. 38, 2018, pp. 370-380, ISSN 1474-0346. https://doi.org/10.1016/j.aei.2018.08.009

Book Chapter

1. **Nizam, R. S.** & Zhang, C. (2017) Industry Foundation Class (IFC) based Data Exchange Model in "Integrated Building Information Modelling" by Bentham Science Publishers. Doi: 10.2174/97816810845721170101

Conference Papers

1. **Nizam, R. S.** & Zhang, C. (2015) Current State of Information Exchange between the two most popular BIM software: Revit and Tekla. International Conference on Sustainable Buildings and Structures, Suzhou, P.R. China.

2. Mehdi, P., **Nizam, R. S.** & Zhang, C. (2015) Investigating thermal comfort and occupants position impacts on building sustainability using CFD and BIM. 49th International Conference of the Architectural Science Association, Australia.

3. **Nizam, R. S.** et. al. (2017) BIM-Based Investigation of Total Energy Consumption in Delivering Building Products. ICCBEI & CCACHE, Taipei, Taiwan. April, 2017.

4. **Nizam, R. S.** & Zhang, C. (2017) Lifecycle Evaluation of Building Sustainability Using BIM. World Sustainable Built Environment Conference 2017 Hong Kong. June, 2017

5. **Nizam, R.S.** et. al. (2017) Simulating total embodied energy of building products through BIM. Winter Simulation Conference (WSC), Las Vegas, NV, 2017, pp. 2394-2404. doi: 10.1109/WSC.2017.8247969.

6. **Nizam, R. S.** & Zhang, C. (2018) BIM Extension to incorporate Embodied Energy information. CONVR2018, Auckland, New Zealand November 2018.

APPENDIX A: Source Code

Source Code for Prototype tool

Initialization:

```
ExtendedCmbBox Cmb2 = new ExtendedCmbBox();
ExtendedCmbBox Cmb1 = new ExtendedCmbBox();
ExtendedBttn B1 = new ExtendedBttn();
        ArrayList arrMaterial = new ArrayList();
        DataTable dtAllwalls = new DataTable();
        DataTable dtAllroofs = new DataTable();
        DataTable dtAllcolumns = new DataTable();
        DataTable dtAllbeams = new DataTable();
        DataTable dtAllfloors = new DataTable();
        DataTable dtAllfoundations = new DataTable();
DataTable dtAllQty = new DataTable();
string connetionString = "Data Source=GAME-PC; Integrated Security=SSPI;" +
"Initial Catalog=SURF2017";
        SqlConnection connection;
        SqlCommand command;
        string AllMaterialName = "";
        int countm = 0;
```

```
  // Calling the functions needed to fill in the EmbEnergy Tabs
            RevitdataWalls();
            RevitdataFloors();
            RevitdataRoofs();
            RevitdataColumns();
           // fillVehicleType();
            RevitdataBeams();
            RevitdataFoundation();
            createdynamicmaterialtabs();
```

6.11.1.1 Dynamic Tabs, Labels and Text Boxes

```
private void createdynamicmaterialtabs()
        {


            dtAllQty.Columns.Add("Qty", typeof(decimal));
```

```
dtAllQty.Columns.Add("Material", typeof(string));


tabControl2.Dock = DockStyle.Left;
tabControl2.BackColor = System.Drawing.Color.White;
tabControl2.ForeColor = System.Drawing.Color.Black;


string[] allmaterials = AllMaterialName.Split(',');



int j = allmaterials.Count();


int counttab = 0;
foreach (string obj in allmaterials)
{
    if (obj != "Air")
    {


        counttab++;

        TabPage tabPage1 = new TabPage();
        tabPage1.Name = "tabPage" + counttab;
        tabPage1.Text = obj.ToString();
        tabPage1.BackColor = System.Drawing.Color.White;
        tabPage1.ForeColor = System.Drawing.Color.Black;
        tabPage1.Font = new Font("Microsoft Sans Serif", 8);
        tabPage1.Width = 1058;
        tabPage1.Height = 394;
        tabControl2.TabPages.Add(tabPage1);


        Label label1 = new Label();
        label1.Text = "Quantity";
        label1.BackColor = System.Drawing.Color.White;
        label1.ForeColor = System.Drawing.Color.Black;
        label1.Font = new Font("Microsoft Sans Serif", 8);
        label1.Width = 61;
        label1.Height = 17;
        label1.Location = new System.Drawing.Point(315, 60);
        tabPage1.Controls.Add(label1);
        label1.Name = obj.ToString() + "Label 1";


        Label label2 = new Label();
        label2.Text = "Unit";
        label2.BackColor = System.Drawing.Color.White;
        label2.ForeColor = System.Drawing.Color.Black;
        label2.Font = new Font("Microsoft Sans Serif", 8);
        label2.Width = 33;
        label2.Height = 17;
        label2.Location = new System.Drawing.Point(492, 61);
        tabPage1.Controls.Add(label2);

        Label label3 = new Label();
        label3.Text = "Density";
        label3.BackColor = System.Drawing.Color.White;
        label3.ForeColor = System.Drawing.Color.Black;
        label3.Font = new Font("Microsoft Sans Serif", 8);
```

```
label3.Width = 55;
label3.Height = 17;
label3.Location = new System.Drawing.Point(653, 60);
tabPage1.Controls.Add(label3);

Label label4 = new Label();
label4.Text = "Vehicle Type";
label4.BackColor = System.Drawing.Color.White;
label4.ForeColor = System.Drawing.Color.Black;
label4.Font = new Font("Microsoft Sans Serif", 8);
label4.Width = 90;
label4.Height = 17;
label4.Location = new System.Drawing.Point(47, 113);
tabPage1.Controls.Add(label4);

Label label5 = new Label();
label5.Text = "Capacity";
label5.BackColor = System.Drawing.Color.White;
label5.ForeColor = System.Drawing.Color.Black;
label5.Font = new Font("Microsoft Sans Serif", 8);
label5.Width = 63;
label5.Height = 17;
label5.Location = new System.Drawing.Point(271, 109);
tabPage1.Controls.Add(label5);

Label label6 = new Label();
label6.Text = "Vehicle English Name";
label6.BackColor = System.Drawing.Color.White;
label6.ForeColor = System.Drawing.Color.Black;
label6.Font = new Font("Microsoft Sans Serif", 8);
label6.Width = 143;
label6.Height = 17;
label6.Location = new System.Drawing.Point(475, 104);
tabPage1.Controls.Add(label6);

Label label7 = new Label();
label7.Text = "Vehicle Chinese Name";
label7.BackColor = System.Drawing.Color.White;
label7.ForeColor = System.Drawing.Color.Black;
label7.Font = new Font("Microsoft Sans Serif", 8);
label7.Width = 143;
label7.Height = 17;
label7.Location = new System.Drawing.Point(730, 104);
tabPage1.Controls.Add(label7);

Label label8 = new Label();
label8.Text = "Fuel Consumption";
label8.BackColor = System.Drawing.Color.White;
label8.ForeColor = System.Drawing.Color.Black;
label8.Font = new Font("Microsoft Sans Serif", 8);
label8.Width = 121;
label8.Height = 17;
label8.Location = new System.Drawing.Point(43, 157);
tabPage1.Controls.Add(label8);

Label label9 = new Label();
label9.Text = "Mass";
label9.BackColor = System.Drawing.Color.White;
label9.ForeColor = System.Drawing.Color.Black;
label9.Font = new Font("Microsoft Sans Serif", 8);
label9.Width = 41;
label9.Height = 17;
```

```
                        label9.Location = new System.Drawing.Point(839, 62);
                        tabPage1.Controls.Add(label9);

                        Label label10 = new Label();
                        label10.Text = "Capacity";
                        label10.BackColor = System.Drawing.Color.White;
                        label10.ForeColor = System.Drawing.Color.Black;
                        label10.Font = new Font("Microsoft Sans Serif", 8);
                        label10.Width = 79;
                        label10.Height = 17;
                        label10.Location = new System.Drawing.Point(507, 157);
                        tabPage1.Controls.Add(label10);

                        Label label11 = new Label();
                        label11.Text = "Hauling Distance";
                        label11.BackColor = System.Drawing.Color.White;
                        label11.ForeColor = System.Drawing.Color.Black;
                        label11.Font = new Font("Microsoft Sans Serif", 8);
                        label11.Width = 115;
                        label11.Height = 17;
                        label11.Location = new System.Drawing.Point(43, 212);
                        tabPage1.Controls.Add(label11);

                        Label label12 = new Label();
                        label12.Text = "Round Trip";
                        label12.BackColor = System.Drawing.Color.White;
                        label12.ForeColor = System.Drawing.Color.Black;
                        label12.Font = new Font("Microsoft Sans Serif", 8);
                        label12.Width = 79;
                        label12.Height = 17;
                        label12.Location = new System.Drawing.Point(507, 212);
                        tabPage1.Controls.Add(label12);

                        Label label13 = new Label();
                        label13.Text = "Fuel Energy Coefficeint";
                        label13.BackColor = System.Drawing.Color.White;
                        label13.ForeColor = System.Drawing.Color.Black;
                        label13.Font = new Font("Microsoft Sans Serif", 8);
                        label13.Width = 154;
                        label13.Height = 17;
                        label13.Location = new System.Drawing.Point(43, 259);
                        tabPage1.Controls.Add(label13);

                        Label label14 = new Label();
                        label14.Text = "Embodied Energy";
                        label14.BackColor = System.Drawing.Color.White;
                        label14.ForeColor = System.Drawing.Color.Black;
                        label14.Font = new Font("Microsoft Sans Serif", 8);
                        label14.Width = 120;
                        label14.Height = 17;
                        label14.Location = new System.Drawing.Point(43, 344);
                        tabPage1.Controls.Add(label14);




  TextBox txtQuantity = new TextBox();
                        txtQuantity.Name    =    Regex.Replace("txtQuantity"    +
obj.ToString(), @"\s+", "");
                        txtQuantity.BackColor = System.Drawing.Color.White;
                        txtQuantity.ForeColor = System.Drawing.Color.Black;
```

```
                    txtQuantity.Font = new Font("Microsoft Sans Serif", 8);
                    txtQuantity.Width = 84;
                    txtQuantity.Height = 22;
                    txtQuantity.Location = new System.Drawing.Point(391, 63);
                    tabPage1.Controls.Add(txtQuantity);

                    TextBox txt2 = new TextBox();
                    txt2.Name  =  Regex.Replace("txtUnit"  +  obj.ToString(),
@"\s+", "");

                    txt2.BackColor = System.Drawing.Color.White;
                    txt2.ForeColor = System.Drawing.Color.Black;
                    txt2.Font = new Font("Microsoft Sans Serif", 8);
                    txt2.Width = 84;
                    txt2.Height = 22;
                    txt2.Location = new System.Drawing.Point(543, 57);
                    tabPage1.Controls.Add(txt2);

                    TextBox txt3 = new TextBox();
                    txt3.Name  =  Regex.Replace("txtdty"  +  obj.ToString(),
@"\s+", "");

                    txt3.BackColor = System.Drawing.Color.White;
                    txt3.ForeColor = System.Drawing.Color.Black;
                    txt3.Font = new Font("Microsoft Sans Serif", 8);
                    txt3.Width = 100;
                    txt3.Height = 22;
                    txt3.Location = new System.Drawing.Point(714, 60);
                    tabPage1.Controls.Add(txt3);

                    TextBox txt4 = new TextBox();
                    txt4.Name  =  Regex.Replace("txtMass"  +  obj.ToString(),
@"\s+", "");

                    txt4.BackColor = System.Drawing.Color.White;
                    txt4.ForeColor = System.Drawing.Color.Black;
                    txt4.Font = new Font("Microsoft Sans Serif", 8);
                    txt4.Width = 78;
                    txt4.Height = 22;
                    txt4.Location = new System.Drawing.Point(899, 57);
                    tabPage1.Controls.Add(txt4);

                    TextBox txt5 = new TextBox();
                    txt5.Name     =     Regex.Replace("txtVEnglishName"     +
obj.ToString(), @"\s+", "");
                    txt5.BackColor = System.Drawing.Color.White;
                    txt5.ForeColor = System.Drawing.Color.Black;
                    txt5.Font = new Font("Microsoft Sans Serif", 8);
                    txt5.Width = 84;
                    txt5.Height = 22;
                    txt5.Location = new System.Drawing.Point(624, 99);
                    tabPage1.Controls.Add(txt5);

                    TextBox txt6 = new TextBox();
                    txt6.Name     =     Regex.Replace("txtVChineseName"     +
obj.ToString(), @"\s+", "");
                    txt6.BackColor = System.Drawing.Color.White;
                    txt6.ForeColor = System.Drawing.Color.Black;
                    txt6.Font = new Font("Microsoft Sans Serif", 8);
                    txt6.Width = 100;
                    txt6.Height = 22;
                    txt6.Location = new System.Drawing.Point(888, 101);
                    tabPage1.Controls.Add(txt6);

                    TextBox txt7 = new TextBox();
```

```
                    txt7.Name    =    Regex.Replace("txtFuelConsumption"    +
obj.ToString(), @"\s+", "");
                    txt7.BackColor = System.Drawing.Color.White;
                    txt7.ForeColor = System.Drawing.Color.Black;
                    txt7.Font = new Font("Microsoft Sans Serif", 8);
                    txt7.Width = 288;
                    txt7.Height = 22;
                    txt7.Location = new System.Drawing.Point(213, 157);
                    tabPage1.Controls.Add(txt7);

                    TextBox txt8 = new TextBox();
                    txt8.Name = Regex.Replace("txtCapacity" + obj.ToString(),
@"\s+", "");
                    txt8.BackColor = System.Drawing.Color.White;
                    txt8.ForeColor = System.Drawing.Color.Black;
                    txt8.Font = new Font("Microsoft Sans Serif", 8);
                    txt8.Width = 288;
                    txt8.Height = 22;
                    txt8.Location = new System.Drawing.Point(689, 157);
                    tabPage1.Controls.Add(txt8);


                    TextBox txt10 = new TextBox();
                    txt10.Name    =    Regex.Replace("txtHaulingDistance"    +
obj.ToString(), @"\s+", "");
                    txt10.BackColor = System.Drawing.Color.White;
                    txt10.ForeColor = System.Drawing.Color.Black;
                    txt10.Font = new Font("Microsoft Sans Serif", 8);
                    txt10.Width = 288;
                    txt10.Height = 22;
                    txt10.Location = new System.Drawing.Point(213, 212);
                    tabPage1.Controls.Add(txt10);

                    TextBox txt11 = new TextBox();
                    txt11.Name     =     Regex.Replace("txtRoundTrip"     +
obj.ToString(), @"\s+", "");
                    txt11.BackColor = System.Drawing.Color.White;
                    txt11.ForeColor = System.Drawing.Color.Black;
                    txt11.Font = new Font("Microsoft Sans Serif", 8);
                    txt11.Width = 288;
                    txt11.Height = 22;
                    txt11.Location = new System.Drawing.Point(689, 212);
                    tabPage1.Controls.Add(txt11);

                    TextBox txt12 = new TextBox();
                    txt12.Name     =     Regex.Replace("txtFuelEnergy"     +
obj.ToString(), @"\s+", "");
                    txt12.BackColor = System.Drawing.Color.White;
                    txt12.ForeColor = System.Drawing.Color.Black;
                    txt12.Font = new Font("Microsoft Sans Serif", 8);
                    txt12.Width = 288;
                    txt12.Height = 22;
                    txt12.Location = new System.Drawing.Point(213, 259);
                    tabPage1.Controls.Add(txt12);

                    TextBox txt13 = new TextBox();
                    txt13.Name = Regex.Replace("textBox5" + obj.ToString(),
@"\s+", "");

                    txt13.BackColor = System.Drawing.Color.White;
                    txt13.ForeColor = System.Drawing.Color.Black;
                    txt13.Font = new Font("Microsoft Sans Serif", 8);
                    txt13.Width = 288;
```

```
                        txt13.Height = 22;
                        txt13.Location = new System.Drawing.Point(213, 339);
                        tabPage1.Controls.Add(txt13);
```

FILL HAULING DISTANCE FUNCTION

```
private decimal fillDistance(string c )
        {
            decimal haul = 0;
            string connetionString = null;
            SqlConnection connection;
            SqlCommand command;
            SqlDataAdapter adapter = new SqlDataAdapter();
            DataSet ds = new DataSet();
            int i = 0;
            string sql = null;
            connetionString    =    "Data    Source=GAME-PC;    Integrated
Security=SSPI;" + "Initial Catalog=SURF2017";

            string material = c.Substring(0, 3);

            sql = "select distance from Transport where Material like  '%" +
material + "%'";
            connection = new SqlConnection(connetionString);

                connection.Open();
                command = new SqlCommand(sql, connection);
                adapter.SelectCommand = command;
                adapter.Fill(ds);
                adapter.Dispose();
                command.Dispose();
                connection.Close();

                foreach (DataRow table in ds.Tables[0].Rows)
                {
                    // txtHaulingDistance.Text = .ToString();
                     haul = decimal.Parse( table["Distance"].ToString());


                }


            return haul;
        }
```

6.11.1.2 Dynamic Combo box for Transportation tab

```
                    Cmb1 = new ExtendedCmbBox();
                    string  cmb1name  =  Regex.Replace("ddlVehicleType"  +
obj.ToString(), @"\s+", "");
                    Cmb1.Name = cmb1name;
                    Cmb1.BackColor = System.Drawing.Color.White;
                    Cmb1.ForeColor = System.Drawing.Color.Black;
                    Cmb1.Font = new Font("Microsoft Sans Serif", 8);
                    Cmb1.Width = 121;
                    Cmb1.Height = 24;
                    Cmb1.Location = new System.Drawing.Point(144, 106);
                    Cmb1._myval = Cmb1.Name;
```

```
                    Cmb1.SelectedIndexChanged             +=             new
EventHandler(Cmb1_SelectedIndexChanged);
                    tabPage1.Controls.Add(Cmb1);


                    Cmb2 = new ExtendedCmbBox();
                    string  cmb2name  =  Regex.Replace("ddlVehicleCode"  +
obj.ToString(), @"\s+", "");
                    Cmb2.Name = cmb2name;
                    Cmb2.BackColor = System.Drawing.Color.White;
                    Cmb2.ForeColor = System.Drawing.Color.Black;
                    Cmb2.Font = new Font("Microsoft Sans Serif", 8);
                    Cmb2.Width = 121;
                    Cmb2.Height = 24;
                    Cmb2.Location = new System.Drawing.Point(340, 104);
                    Cmb2.SelectedIndexChanged             +=             new
EventHandler(Cmb2_SelectedIndexChanged);
                    tabPage1.Controls.Add(Cmb2);

                    fillVehicleType(Cmb1, Cmb2);
```

6.11.1.3 Dynamic Buttons

```
                    B1 = new ExtendedBttn();
                    B1.Name = "btnCalTenergy" + obj.ToString();
                    B1.Text = "Calculate T.E. for " + obj.ToString();
                    B1.BackColor = System.Drawing.Color.White;
                    B1.ForeColor = System.Drawing.Color.Black;
                    B1.Font = new Font("Microsoft Sans Serif", 8);
                    B1.Width = 248;
                    B1.Height = 23;
                    B1.Location = new System.Drawing.Point(213, 298);
                    tabPage1.Controls.Add(B1);
                    B1.Click += new EventHandler(B1_Click);

                }

}
```

6.11.1.4 Button Click Event

```
   private void B1_Click(object sender, EventArgs e)
        {


            decimal FEC = 0;
            decimal FConsmp = 0;
            int trips = 0;
            decimal haul = 0;
            string Bname = ((ExtendedBttn)sender).Name;

            //string container= ((ExtendedBttn)sender).Container.ToString();
```

```
            foreach (System.Windows.Forms.Control c in tabControl2.Controls)
            {
                string FuelConsumption = Regex.Replace("txtFuelConsumption"
+ c.Text, @"\s+", "");
                string FuelEnergy = Regex.Replace("txtFuelEnergy" + c.Text,
@"\s+", "");
                string Haul = Regex.Replace("txtHaulingDistance" + c.Text,
@"\s+", "");
                string Trip = Regex.Replace("txtRoundTrip" + c.Text, @"\s+",
"");




                if (c is TabPage && c.Text.Contains("gray"))
                    foreach (System.Windows.Forms.TextBox childc in
c.Controls.OfType<TextBox>())

                    {

                        if (childc.Name == FuelConsumption )
                        {

                            FConsmp = decimal.Parse(childc.Text);
                            MessageBox.Show("Fconmp: " + FConsmp);
                        }
                        else if (childc.Name == FuelEnergy)
                        {

                            FEC = decimal.Parse(childc.Text);
                            MessageBox.Show("FEC" + FEC);
                        }
                        else if (childc.Name == Haul)
                        {

                            haul = decimal.Parse(childc.Text);
                            MessageBox.Show("haul :" + haul);
                        }
                        else if (childc.Name == Trip)
                        {

                            trips = int.Parse(childc.Text);
                            MessageBox.Show("trips" + trips);
                        }



                    }


            }
            foreach (System.Windows.Forms.Control c in tabControl2.Controls)
            {
                string EmbEnrgy = Regex.Replace("textBox5" + c.Text, @"\s+",
"");

                if (c is TabPage && c.Text.Contains("gray"))
                {
```

```
                    foreach    (System.Windows.Forms.TextBox    childc    in
c.Controls.OfType<TextBox>())
                    {
                        if (childc.Name == EmbEnrgy)
                        {
                            MessageBox.Show("haul  :"  + haul  + "FEC"+ FEC+
"trips" + trips+ "Fconmp: " + FConsmp);
                            decimal energy = haul * 2 / 100 * FEC * trips *
FConsmp;
                            childc.Text = energy.ToString();

                        }

                    }




                }





            }
        }
```

6.11.1.5 Vehicle Type Combo Box

```
public  void  fillVehicleType(  ComboBox  ddlVehicleTypeMaterial,  ComboBox
ddlVehicleCodeMaterial)
        {

            string connetionString = null;
            SqlConnection connection;
            SqlCommand command;
            SqlDataAdapter adapter = new SqlDataAdapter();
            DataSet ds = new DataSet();
            int i = 0;
            string sql = null;
            connetionString    =    "Data    Source=GAME-PC;    Integrated
Security=SSPI;" + "Initial Catalog=SURF2017";
            sql = "select distinct type from vehicle";
            connection = new SqlConnection(connetionString);
            try
            {
                connection.Open();
                command = new SqlCommand(sql, connection);
                adapter.SelectCommand = command;
                adapter.Fill(ds);
                adapter.Dispose();
                command.Dispose();
                connection.Close();


                ddlVehicleTypeMaterial.DataSource = ds.Tables[0];
                ddlVehicleTypeMaterial.ValueMember = "type";
```

```
                ddlVehicleTypeMaterial.DisplayMember = "type";

            fillVehicleCode(ddlVehicleCodeMaterial,
ddlVehicleTypeMaterial.SelectedValue.ToString());


            }
            catch (Exception ex)
            {
                MessageBox.Show("Can not open connection fillVehicleType !
");
            }
        }
```

CHANGE EVENT

```
private void Cmb1_SelectedIndexChanged(object sender, EventArgs e)
        {
            string CBname = ((ExtendedCmbBox)sender).Name;

            int selectedndex = ((ExtendedCmbBox)sender).SelectedIndex;
            string                      selectedValue                      =
((ExtendedCmbBox)sender).SelectedValue.ToString();



                foreach       (System.Windows.Forms.Control       c       in
tabControl2.Controls)
            {
                    foreach    (System.Windows.Forms.ComboBox    childc    in
c.Controls.OfType<ComboBox>())


                {
                        if (childc.Name == CBname)
                        {
                        string VCddlname = Regex.Replace("ddlVehicleCode" +
childc.Parent.Text, @"\s+", "");

                        foreach   (System.Windows.Forms.ComboBox   childd   in
c.Controls.OfType<ComboBox>())
                        {
                            if (childd.Name == VCddlname)
                            {
                                fillVehicleCode(childd, selectedValue);
                            }

                        }


                    }

                    }
                }

        }
```

6.11.1.6 Vehicle Capacity ComboBox

```
public    void    fillVehicleCode(ComboBox    ddlVehicleCodeMaterial,string
selectedValue)
        {


            string connetionString = null;
            SqlConnection connection;
            SqlCommand command;
            SqlDataAdapter adapter = new SqlDataAdapter();
            DataSet ds = new DataSet();
            int i = 0;
            string sql = null;
            // string type = ddlVehicleType.SelectedValue.ToString();
            //string type = Cmb1.SelectedValue.ToString();
            string type = selectedValue;

            connetionString    =    "Data    Source=GAME-PC;    Integrated
Security=SSPI;" + "Initial Catalog=SURF2017";
            sql = "select Capacity from vehicle where type ='" + type + "'";
            connection = new SqlConnection(connetionString);
            try
            {
                connection.Open();
                command = new SqlCommand(sql, connection);
                adapter.SelectCommand = command;
                adapter.Fill(ds);
                adapter.Dispose();
                command.Dispose();
                connection.Close();


                ddlVehicleCodeMaterial.DataSource = ds.Tables[0];
                ddlVehicleCodeMaterial.ValueMember = "Capacity";
                ddlVehicleCodeMaterial.DisplayMember = "Capacity";

            }
            catch (Exception ex)
            {
                MessageBox.Show("Can  not  open  connection  fillVehicleCode  !
");
            }
        }
```

CHANGE EVENT

```
private void Cmb2_SelectedIndexChanged(object sender, EventArgs e)
        {

            string CBname = ((ExtendedCmbBox)sender).Name;

            int selectedndex = ((ExtendedCmbBox)sender).SelectedIndex;
            string                      selectedValue                      =
((ExtendedCmbBox)sender).SelectedValue.ToString();



            foreach (System.Windows.Forms.Control c in tabControl2.Controls)
            {
                foreach     (System.Windows.Forms.ComboBox     childc     in
c.Controls.OfType<ComboBox>())
```

```
                    {
                        if (childc.Name == CBname)
                        {
                            string VCddlname = Regex.Replace("ddlVehicleType" +
childc.Parent.Text, @"\s+", "");

                            foreach  (System.Windows.Forms.ComboBox  childd  in
c.Controls.OfType<ComboBox>())
                            {
                                if (childd.Name == VCddlname)
                                {


GetVehicleParameters(childd.SelectedValue.ToString(),selectedValue,
childc.Parent.Text);
                                }

                            }


                        }

                    }
                }

            }
```

6.11.1.7 Vehicle Parameters

```
     public void GetVehicleParameters(string type, string Code,string tabname)
        {
            string connetionString = null;
            SqlConnection connection;
            SqlCommand command;
            SqlDataAdapter adapter = new SqlDataAdapter();
            DataSet ds = new DataSet();
            int i = 0;
            string sql = null;
            connetionString    =    "Data    Source=GAME-PC;    Integrated
Security=SSPI;" + "Initial Catalog=SURF2017";


            sql  =  "select  *  from  Vehicle  where  type  =  '"  +  type  +  "'  and
Capacity = '" + Code + "'";
            connection = new SqlConnection(connetionString);
            try
            {
                connection.Open();
                command = new SqlCommand(sql, connection);
                adapter.SelectCommand = command;
                adapter.Fill(ds);
                adapter.Dispose();
                command.Dispose();
                connection.Close();



                string FuelConsumption = Regex.Replace("txtFuelConsumption"
+ tabname, @"\s+", "");
```

```
                string  Capacity  =  Regex.Replace("txtCapacity"  +  tabname,
@"\s+", "");
                string   EnglishName   =   Regex.Replace("txtVEnglishName"   +
tabname, @"\s+", "");
                string   ChineseName   =   Regex.Replace("txtVChineseName"   +
tabname, @"\s+", "");
                string FuelEnergy = Regex.Replace("txtFuelEnergy" + tabname,
@"\s+", "");
                string  Quantity  =  Regex.Replace("txtQuantity"  +  tabname,
@"\s+", "");
                string DENSITY = Regex.Replace("txtdty" + tabname, @"\s+",
"");
                string Mass = Regex.Replace("txtMass" + tabname, @"\s+", "");



                foreach       (System.Windows.Forms.Control       c       in
tabControl2.Controls)
                {
                    foreach     (System.Windows.Forms.TextBox     childc     in
c.Controls.OfType<TextBox>())


                    {
                        if (childc.Name == FuelConsumption)
                        {
                            childc.Text                                      =
ds.Tables[0].Rows[0]["FuelConsumption"].ToString();
                        }
                        else if (childc.Name == Capacity)
                        {
                            childc.Text                                      =
ds.Tables[0].Rows[0]["Capacity"].ToString();
                        }
                        else if (childc.Name == EnglishName)
                        {
                            childc.Text                                      =
ds.Tables[0].Rows[0]["EnglishName"].ToString();
                        }
                        else if (childc.Name == ChineseName)
                        {
                            childc.Text                                      =
ds.Tables[0].Rows[0]["ChineseName"].ToString();
                        }
                        else if (childc.Name == FuelEnergy)
                        {
                            childc.Text                                      =
ds.Tables[0].Rows[0]["LowerHeating"].ToString();
                        }
                        else if (childc.Name == Quantity)
                        {
                            decimal mQty = 0;

                            foreach (DataRow qty in dtAllQty.Rows)
                            {
                                decimal Q = decimal.Parse(qty[0].ToString());
                                string mName = qty[1].ToString();

                                if (mName == tabname)
                                {
```

```
                            mQty = mQty + Q;
                        }


                    }
                    childc.Text = mQty.ToString();
                }




            }



        }


    }
    catch (Exception ex)
    {

    }
}
```

6.11.1.8 Walls

MATERIAL COMBOBOX

```
private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
        {
            txtEECoefficient.Text = "";
            txtDensity.Text = "";
            txtEName.Text = "";
            txtCName.Text = "";
            txtCode.Text = "";

            double volume = 0;
            DataRow                    selectedDataRow                    =
((DataRowView)comboBox2.SelectedItem).Row;
            int dtmid = Convert.ToInt32(selectedDataRow["Id"]);
            string materialname = selectedDataRow["Material"].ToString();


            if (dtmid != null)
            {


                DataTable dtElement = new DataTable();
                dtElement.Columns.Add("ElementId", typeof(string));

                foreach (DataRow tableall in dtAllwalls.Rows)
                {
                    int? dtallmid = int.Parse(tableall[0].ToString());

                    if (dtmid == dtallmid)
```

```
                    {
                        dtElement.Rows.Add(tableall[3].ToString());
                    }

                }

                foreach (DataRow edata in dtElement.Rows)
                {

                    IList<Element> m_elementsToProcess;
                    FilteredElementCollector      collector      =      new
FilteredElementCollector(m_doc);
                    m_elementsToProcess                                   =
collector.OfCategory(BuiltInCategory.OST_Walls).WhereElementIsNotElementTyp
e().ToElements();
                    foreach (Element e1 in m_elementsToProcess)
                    {

                        ICollection<ElementId>         materials          =
e1.GetMaterialIds(false);
                        foreach (ElementId materialId in materials)
                        {

                            ElementId elementId = e1.Id;
                            if (materialname.Contains("Concrete"))
                            {
                                Material              material              =
m_doc.GetElement(materialId) as Material;
                                string SAId = "";
                                string strength = "";
                                SAId = material.StructuralAssetId.ToString();
                                if (SAId != null && SAId != "-1")
                                {
                                    PropertySetElement      property      =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                                    Parameter   paramConcreteCompression   =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);

                                    strength                              =
paramConcreteCompression.AsValueString();
                                    txtSubType.Text = strength;

                                    fillEEFields(materialname, "walls");
                                }
                            }

                            string eid = edata[0].ToString();
                            if (dtmid.ToString() == materialId.ToString())
                            {
                                string e1id = e1.Id.ToString();
                                if (eid == e1id)
                                {
                                    volume        =        volume        +
e1.GetMaterialVolume(materialId);
                                }
                            }

                        }
```

```
                }
            }
            volume = 0.0283168467117 * volume;
            txtQty.Text = volume.ToString();

        }
    }
```

CALCULATE BUTTON CODE

```
private void btnCalEachWall_Click(object sender, EventArgs e)
        {
            decimal qty = decimal.Parse(txtQty.Text);
            decimal density = decimal.Parse(txtDensity.Text);
            decimal EEco = decimal.Parse(txtEECoefficient.Text);
            decimal EE = qty * density * EEco;

            DataRow                     selectedDataRow                    =
((DataRowView)comboBox2.SelectedItem).Row;
            string Materialname = selectedDataRow["Material"].ToString();
            lbWall1.Items.Add(Materialname);
            lbWall2.Items.Add(EE.ToString());
        }
```

6.11.1.9 Floors

MATERIAL COMBOBOX

```
  private    void    ddlMaterialFloor_SelectedIndexChanged(object    sender,
EventArgs e)
        {
            txtEECoefficientFloor.Text = "";
            txtDensityFloor.Text = "";
            txtENameFloor.Text = "";
            txtCNameFloor.Text = "";
            txtCodeFloor.Text = "";

            double volume = 0;
            DataRow                     selectedDataRow                    =
((DataRowView)ddlMaterialFloor.SelectedItem).Row;
            int dtmid = Convert.ToInt32(selectedDataRow["Id"]);
            string materialname = selectedDataRow["Material"].ToString();


            if (dtmid != null)
            {


                DataTable dtElement = new DataTable();
                dtElement.Columns.Add("ElementId", typeof(string));

                foreach (DataRow tableall in dtAllfloors.Rows)
                {
                    int? dtallmid = int.Parse(tableall[0].ToString());

                    if (dtmid == dtallmid)
```

```
                    {
                        dtElement.Rows.Add(tableall[3].ToString());
                    }

                }

                foreach (DataRow edata in dtElement.Rows)
                {

                    IList<Element> m_elementsToProcess;
                    FilteredElementCollector      collector      =      new
FilteredElementCollector(m_doc);
                    m_elementsToProcess                                   =
collector.OfCategory(BuiltInCategory.OST_Floors).WhereElementIsNotElementTy
pe().ToElements();
                    foreach (Element e1 in m_elementsToProcess)
                    {

                        ICollection<ElementId>          materials          =
e1.GetMaterialIds(false);
                        foreach (ElementId materialId in materials)
                        {

                            ElementId elementId = e1.Id;
                            if (materialname.Contains("Concrete"))
                            {
                                Material              material              =
m_doc.GetElement(materialId) as Material;
                                string SAId = "";
                                string strength = "";
                                SAId = material.StructuralAssetId.ToString();
                                if (SAId != null && SAId != "-1")
                                {
                                    PropertySetElement       property       =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                                    Parameter   paramConcreteCompression   =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);

                                    strength                               =
paramConcreteCompression.AsValueString();
                                    txtSubTypeFloor.Text = strength;

                                    fillEEFields(materialname, "floors");

                                }
                            }

                            string eid = edata[0].ToString();
                            if (dtmid.ToString() == materialId.ToString())
                            {
                                string e1id = e1.Id.ToString();
                                if (eid == e1id)
                                {
                                    volume        =         volume         +
e1.GetMaterialVolume(materialId);
                                }
                            }

                        }
```

```
                }
            }
            volume = 0.0283168467117 * volume;
            txtQtyFloor.Text = volume.ToString();

        }
    }
```

CALCULATE BUTTON CODE

```
private void btnCalEachFloor_Click(object sender, EventArgs e)
        {
            decimal qty = decimal.Parse(txtQtyFloor.Text);
            decimal density = decimal.Parse(txtDensityFloor.Text);
            decimal EEco = decimal.Parse(txtEECoefficientFloor.Text);
            decimal EE = qty * density * EEco;

            DataRow                     selectedDataRow                    =
((DataRowView)ddlMaterialFloor.SelectedItem).Row;
            string Materialname = selectedDataRow["Material"].ToString();
            lbFloor1.Items.Add(Materialname);
            lbFloor2.Items.Add(EE.ToString());
        }
```

6.11.1.10 Roofs

MATERIAL COMBOBOX

```
private void ddlMaterialRoof_SelectedIndexChanged(object sender, EventArgs e)
        {
            txtEECoefficientRoof.Text = "";
            txtDensityRoof.Text = "";
            txtENameRoof.Text = "";
            txtCNameRoof.Text = "";
            txtCodeRoof.Text = "";
            double volume = 0;
            DataRow                     selectedDataRow                    =
((DataRowView)ddlMaterialRoof.SelectedItem).Row;
            int dtmid = Convert.ToInt32(selectedDataRow["Id"]);
            string materialname = selectedDataRow["Material"].ToString();


            if (dtmid != null)
            {


                DataTable dtElement = new DataTable();
                dtElement.Columns.Add("ElementId", typeof(string));

                foreach (DataRow tableall in dtAllroofs.Rows)
                {
                    int? dtallmid = int.Parse(tableall[0].ToString());

                    if (dtmid == dtallmid)
                    {

                        dtElement.Rows.Add(tableall[3].ToString());
                    }
```

```
                }

                foreach (DataRow edata in dtElement.Rows)
                {

                    IList<Element> m_elementsToProcess;
                    FilteredElementCollector collector = new
FilteredElementCollector(m_doc);
                    m_elementsToProcess =
collector.OfCategory(BuiltInCategory.OST_Roofs).WhereElementIsNotElementTyp
e().ToElements();
                    foreach (Element e1 in m_elementsToProcess)
                    {

                        ICollection<ElementId> materials =
e1.GetMaterialIds(false);
                        foreach (ElementId materialId in materials)
                        {

                            ElementId elementId = e1.Id;
                            if (materialname.Contains("Concrete"))
                            {
                                Material material =
m_doc.GetElement(materialId) as Material;
                                string SAId = "";
                                string strength = "";
                                SAId = material.StructuralAssetId.ToString();
                                if (SAId != null && SAId != "-1")
                                {
                                    PropertySetElement property =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                                    Parameter paramConcreteCompression =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);

                                    strength =
paramConcreteCompression.AsValueString();
                                    txtSubTypeRoof.Text = strength;

                                    fillEEFields(materialname, "roofs");

                                }
                            }

                            string eid = edata[0].ToString();
                            if (dtmid.ToString() == materialId.ToString())
                            {
                                string e1id = e1.Id.ToString();
                                if (eid == e1id)
                                {
                                    volume = volume +
e1.GetMaterialVolume(materialId);
                                }
                            }

                        }

                    }
                }
                volume = 0.0283168467117 * volume;
```

```
                txtQtyRoof.Text = volume.ToString();

            }

        }
```

CALCULATE BUTTON CODE

```
private void btnCalEachRoof_Click(object sender, EventArgs e)
        {
            decimal qty = decimal.Parse(txtQtyRoof.Text);
            decimal density = decimal.Parse(txtDensityRoof.Text);
            decimal EEco = decimal.Parse(txtEECoefficientRoof.Text);
            decimal EE = qty * density * EEco;

            DataRow                     selectedDataRow                    =
((DataRowView)ddlMaterialRoof.SelectedItem).Row;
            string Materialname = selectedDataRow["Material"].ToString();
            lbRoof1.Items.Add(Materialname);
            lbRoof2.Items.Add(EE.ToString());
        }
```

6.11.1.11 Columns

MATERIAL COMBOBOX

```
private void ddlMaterialColumn_SelectedIndexChanged(object sender, EventArgs
e)
        {
            txtEECoefficientColumn.Text = "";
            txtDensityColumn.Text = "";
            txtENameColumn.Text = "";
            txtCNameColumn.Text = "";
            txtCodeColumn.Text = "";
            double volume = 0;
            DataRow                     selectedDataRow                    =
((DataRowView)ddlMaterialColumn.SelectedItem).Row;
            int dtmid = Convert.ToInt32(selectedDataRow["Id"]);
            string materialname = selectedDataRow["Material"].ToString();


            if (dtmid != null)
            {


                DataTable dtElement = new DataTable();
                dtElement.Columns.Add("ElementId", typeof(string));

                foreach (DataRow tableall in dtAllcolumns.Rows)
                {
                    int? dtallmid = int.Parse(tableall[0].ToString());

                    if (dtmid == dtallmid)
                    {

                        dtElement.Rows.Add(tableall[3].ToString());
                    }

                }
```

```
                    foreach (DataRow edata in dtElement.Rows)
                    {

                        IList<Element> m_elementsToProcess;
                        FilteredElementCollector       collector       =       new 
FilteredElementCollector(m_doc);
                        m_elementsToProcess                                    = 
collector.OfCategory(BuiltInCategory.OST_StructuralColumns).WhereElementIsN
otElementType().ToElements();
                        foreach (Element e1 in m_elementsToProcess)
                        {

                            ICollection<ElementId>          materials          = 
e1.GetMaterialIds(false);
                            foreach (ElementId materialId in materials)
                            {

                                ElementId elementId = e1.Id;
                                if (materialname.Contains("Concrete"))
                                {
                                    Material              material              = 
m_doc.GetElement(materialId) as Material;
                                    string SAId = "";
                                    string strength = "";
                                    SAId = material.StructuralAssetId.ToString();
                                    if (SAId != null && SAId != "-1")
                                    {
                                        try
                                        {
                                            PropertySetElement     property     = 
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                                            Parameter paramConcreteCompression = 
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);

                                            strength                            = 
paramConcreteCompression.AsValueString();
                                            txtSubTypeColumn.Text = strength;

                                            fillEEFields(materialname, 
"columns");
                                        }
                                        catch
                                        {


                                        }

                                    }
                                }

                                string eid = edata[0].ToString();
                                if (dtmid.ToString() == materialId.ToString())
                                {
                                    string e1id = e1.Id.ToString();
                                    if (eid == e1id)
                                    {
                                        volume          =          volume          + 
e1.GetMaterialVolume(materialId);
                                    }
```

```
                    }
                }
            }
        }
        volume = 0.0283168467117 * volume;
        txtQtyColumn.Text = volume.ToString();

    }
}
```

CALCULATE BUTTON CODE

```
private void btnCalEachColumn_Click(object sender, EventArgs e)
        {
            decimal qty = decimal.Parse(txtQtyColumn.Text);
            decimal density = decimal.Parse(txtDensityColumn.Text);
            decimal EEco = decimal.Parse(txtEECoefficientColumn.Text);
            decimal EE = qty * density * EEco;

            DataRow                    selectedDataRow                    =
((DataRowView)ddlMaterialColumn.SelectedItem).Row;
            string Materialname = selectedDataRow["Material"].ToString();
            lbColumn1.Items.Add(Materialname);
            lbColumn2.Items.Add(EE.ToString());
        }
```

6.11.1.12 Beams

MATERIAL COMBOBOX

```
private void ddlMaterialBeam_SelectedIndexChanged(object sender, EventArgs e)
        {
            txtEECoefficientBeam.Text = "";
            txtDensityBeam.Text = "";
            txtENameBeam.Text = "";
            txtCNameBeam.Text = "";
            txtCodeBeam.Text = "";
            double volume = 0;
            DataRow                    selectedDataRow                    =
((DataRowView)ddlMaterialBeam.SelectedItem).Row;
            int dtmid = Convert.ToInt32(selectedDataRow["Id"]);
            string materialname = selectedDataRow["Material"].ToString();
            if (dtmid != null)
            {
                DataTable dtElement = new DataTable();
                dtElement.Columns.Add("ElementId", typeof(string));

                foreach (DataRow tableall in dtAllbeams.Rows)
                {
                    int? dtallmid = int.Parse(tableall[0].ToString());

                    if (dtmid == dtallmid)
                    {

                        dtElement.Rows.Add(tableall[3].ToString());
                    }
                }
```

```
                foreach (DataRow edata in dtElement.Rows)
                {

                    ElementClassFilter      beamClassFilter      =      new
ElementClassFilter(typeof(FamilyInstance));

                    ElementCategoryFilter      beamTypeFilter      =      new
ElementCategoryFilter(BuiltInCategory.OST_StructuralFraming);

                    LogicalAndFilter         beamFilter         =         new
LogicalAndFilter(beamClassFilter, beamTypeFilter);

                    IEnumerable<Element> elems = from elem in
                                        new
FilteredElementCollector(m_doc).WherePasses(beamFilter).ToElements()

                                        select elem;
                    foreach (Element e1 in elems)
                    {

                        ICollection<ElementId>         materials          =
e1.GetMaterialIds(false);
                        foreach (ElementId materialId in materials)
                        {
                            ElementId elementId = e1.Id;
                            if (materialname.Contains("Concrete"))
                            {
                                Material              material              =
m_doc.GetElement(materialId) as Material;
                                string SAId = "";
                                string strength = "";
                                SAId = material.StructuralAssetId.ToString();
                                if (SAId != null && SAId != "-1")
                                {
                                    try
                                    {
                                        PropertySetElement     property     =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;
                                        Parameter paramConcreteCompression =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);
                                        strength                          =
paramConcreteCompression.AsValueString();
                                        txtSubTypeBeam.Text = strength;
                                        fillEEFields(materialname, "beams");
                                    }
                                    catch
                                    {
                                    }
                                }
                            }

                            string eid = edata[0].ToString();
                            if (dtmid.ToString() == materialId.ToString())
                            {
                                string e1id = e1.Id.ToString();
                                if (eid == e1id)
                                {
                                    volume        =        volume        +
e1.GetMaterialVolume(materialId);
```

```
                        }
                    }
                }
            }
        }
        volume = 0.0283168467117 * volume;
        txtQtyFloorBeam.Text = volume.ToString();
    }
}
```

CALCULATE BUTTON CODE

```
private void btnCalEachBeam_Click(object sender, EventArgs e)
        {
            decimal qty = decimal.Parse(txtQtyFloorBeam.Text);
            decimal density = decimal.Parse(txtDensityBeam.Text);
            decimal EEco = decimal.Parse(txtEECoefficientBeam.Text);
            decimal EE = qty * density * EEco;

            DataRow                    selectedDataRow                    =
((DataRowView)ddlMaterialBeam.SelectedItem).Row;
            string Materialname = selectedDataRow["Material"].ToString();
            lbBeam1.Items.Add(Materialname);
            lbBeam2.Items.Add(EE.ToString());
        }
```

6.11.1.13 Foundation

MATERIAL COMBOBOX

```
private   void   ddlMaterialFoundation_SelectedIndexChanged(object   sender,
EventArgs e)
        {
            txtEECoefficientFoundation.Text = "";
            txtDensityFoundation.Text = "";
            txtENameFoundation.Text = "";
            txtCNameFoundation.Text = "";
            txtCodeFoundation.Text = "";
            double volume = 0;
            DataRow                    selectedDataRow                    =
((DataRowView)ddlMaterialFoundation.SelectedItem).Row;
            int dtmid = Convert.ToInt32(selectedDataRow["Id"]);
            string materialname = selectedDataRow["Material"].ToString();


            if (dtmid != null)
            {


                DataTable dtElement = new DataTable();
                dtElement.Columns.Add("ElementId", typeof(string));

                foreach (DataRow tableall in dtAllfoundations.Rows)
```

```
                {
                    int? dtallmid = int.Parse(tableall[0].ToString());

                    if (dtmid == dtallmid)
                    {

                        dtElement.Rows.Add(tableall[3].ToString());
                    }

                }

                foreach (DataRow edata in dtElement.Rows)
                {

                    IList<Element> m_elementsToProcess;
                    FilteredElementCollector      collector      =      new
FilteredElementCollector(m_doc);
                    m_elementsToProcess                                   =
collector.OfCategory(BuiltInCategory.OST_StructuralFoundation).WhereElement
IsNotElementType().ToElements();
                    foreach (Element e1 in m_elementsToProcess)
                    {

                        ICollection<ElementId>          materials          =
e1.GetMaterialIds(false);
                        foreach (ElementId materialId in materials)
                        {

                            ElementId elementId = e1.Id;
                            if (materialname.Contains("Concrete"))
                            {
                                Material              material              =
m_doc.GetElement(materialId) as Material;
                                string SAId = "";
                                string strength = "";
                                SAId = material.StructuralAssetId.ToString();
                                if (SAId != null && SAId != "-1")
                                {
                                    try
                                    {
                                        PropertySetElement     property     =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                                        Parameter paramConcreteCompression =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);

                                        strength                            =
paramConcreteCompression.AsValueString();
                                        txtSubTypeFoundation.Text = strength;

                                        fillEEFields(materialname,
"foundations");
                                    }
                                    catch
                                    {


                                    }

                                }
                            }
```

```
                            string eid = edata[0].ToString();
                            if (dtmid.ToString() == materialId.ToString())
                            {
                                string e1id = e1.Id.ToString();
                                if (eid == e1id)
                                {
                                    volume        =        volume        +
e1.GetMaterialVolume(materialId);
                                }
                            }

                        }

                    }
                }
                volume = 0.0283168467117 * volume;
                txtQtyFoundation.Text = volume.ToString();

            }

        }
```

CALCULATE BUTTON CODE

```
private void btnCalEachFoundation_Click(object sender, EventArgs e)
        {
            decimal qty = decimal.Parse(txtQtyFoundation.Text);
            decimal density = decimal.Parse(txtDensityFoundation.Text);
            decimal EEco = decimal.Parse(txtEECoefficientFoundation.Text);
            decimal EE = qty * density * EEco;

            DataRow                    selectedDataRow                    =
((DataRowView)ddlMaterialFoundation.SelectedItem).Row;
            string Materialname = selectedDataRow["Material"].ToString();
            lbFoundation1.Items.Add(Materialname);
            lbFoundation2.Items.Add(EE.ToString());
        }
```

6.11.1.14 Dynamic Total Energy tab

```
TabPage tabPagex = new TabPage();
            tabPagex.Name = "Total Energy";
            tabPagex.Text = "Total Energy";
            tabPagex.BackColor = System.Drawing.Color.White;
            tabPagex.ForeColor = System.Drawing.Color.Black;
            tabPagex.Font = new Font("Microsoft Sans Serif", 8);
            tabPagex.Width = 1058;
            tabPagex.Height = 394;
            Beam.TabPages.Add(tabPagex);

            ListBox list1 = new ListBox();
            list1.Text = "Quantity";
```

```
list1.BackColor = System.Drawing.Color.White;
list1.ForeColor = System.Drawing.Color.Black;
list1.Font = new Font("Microsoft Sans Serif", 8);
list1.Width = 61;
list1.Height = 300;
list1.Location = new System.Drawing.Point(20, 60);
tabPagex.Controls.Add(list1);
list1.Name = "Material";

ListBox list3 = new ListBox();
list3.Text = "Quantity";
list3.BackColor = System.Drawing.Color.White;
list3.ForeColor = System.Drawing.Color.Black;
list3.Font = new Font("Microsoft Sans Serif", 8);
list3.Width = 61;
list3.Height = 300;
list3.Location = new System.Drawing.Point(300, 60);
tabPagex.Controls.Add(list3);
list3.Name = "Element";


ListBox list2 = new ListBox();
list2.Text = "Quantity";
list2.BackColor = System.Drawing.Color.White;
list2.ForeColor = System.Drawing.Color.Black;
list2.Font = new Font("Microsoft Sans Serif", 8);
list2.Width = 61;
list2.Height = 300;
list2.Location = new System.Drawing.Point(200, 60);
tabPagex.Controls.Add(list2);
list2.Name = "Energy";


foreach (string obj in allmaterials)
{
    decimal wall;
    decimal floor;
    decimal foundation;
    decimal roof;
    decimal column;
    decimal beam;

    int indexw = comboBox2.FindString(obj);
    int indexr = ddlMaterialRoof.FindString(obj);
    int indexc = ddlMaterialColumn.FindString(obj);
    int indexb = ddlMaterialBeam.FindString(obj);
    int indexf = ddlMaterialFoundation.FindString(obj);
    int indexfl = ddlMaterialFloor.FindString(obj);



    if (indexfl != -1)
    {
        ddlMaterialFloor.SelectedIndex = indexfl;
        list1.Items.Add("Floor");
        list2.Items.Add(txtQtyFloor.Text);
        list3.Items.Add(obj);
        dtAllQty.Rows.Add(decimal.Parse(txtQtyFloor.Text), obj);

    }
    if (indexw != -1)
    {
```

```
                comboBox2.SelectedIndex = indexw;
                list1.Items.Add("Wall");
                list2.Items.Add(txtQty.Text);
                list3.Items.Add(obj);
            }
            if (indexc != -1)
            {
                ddlMaterialColumn.SelectedIndex = indexc;
                list1.Items.Add("Column");
                list2.Items.Add(txtQtyColumn.Text);
                list3.Items.Add(obj);
                dtAllQty.Rows.Add(decimal.Parse(txtQtyColumn.Text), obj);
            }
            if (indexb != -1)
            {
                ddlMaterialBeam.SelectedIndex = indexb;
                list1.Items.Add("Beam");
                list2.Items.Add(txtQtyFloorBeam.Text);
                list3.Items.Add(obj);
                dtAllQty.Rows.Add(decimal.Parse(txtQtyFloorBeam.Text),
obj);
            }
            if (indexf != -1)
            {
                ddlMaterialFoundation.SelectedIndex = indexf;
                list1.Items.Add("Foundation");
                list2.Items.Add(txtQtyFoundation.Text);
                list3.Items.Add(obj);
                dtAllQty.Rows.Add(decimal.Parse(txtQtyFoundation.Text),
obj);

            }
            if (indexr != -1)
            {
                ddlMaterialRoof.SelectedIndex = indexr;
                list1.Items.Add("Roof");
                list2.Items.Add(txtQtyRoof.Text);
                list3.Items.Add(obj);
                dtAllQty.Rows.Add(decimal.Parse(txtQtyRoof.Text), obj);
            }


        }
        Cmb2.SelectedIndexChanged                 +=                 new
EventHandler(Cmb2_SelectedIndexChanged);
        Cmb2_SelectedIndexChanged(Cmb2, EventArgs.Empty);
        foreach (System.Windows.Forms.Control c in tabControl2.Controls)
        {
            foreach     (System.Windows.Forms.ComboBox     childc     in
c.Controls.OfType<ComboBox>())


            {
                Cmb2_SelectedIndexChanged(childc, EventArgs.Empty);
            }
        }
        decimal qtydecimal = 0;
        decimal dty = 0;
        decimal mass = 0;
        decimal dist = 0;
        decimal capacity = 0;
        int trips = 0;
```

```
            foreach (System.Windows.Forms.Control c in tabControl2.Controls)
            {
                string  Quantity  =  Regex.Replace("txtQuantity"  +  c.Text,
@"\s+", "");
                string DENSITY = Regex.Replace("txtdty" + c.Text, @"\s+", "");
                string Mass = Regex.Replace("txtMass" + c.Text, @"\s+", "");
                string  Haul  =  Regex.Replace("txtHaulingDistance"  +  c.Text,
@"\s+", "");
                string Trip = Regex.Replace("txtRoundTrip" + c.Text, @"\s+",
"");
                string  Capacity  =  Regex.Replace("txtCapacity"  +  c.Text,
@"\s+", "");

                if (c is TabPage && c.Text.Contains("Concrete"))
                {

                    foreach    (System.Windows.Forms.TextBox    childc    in
c.Controls.OfType<TextBox>())
                    {


                        if (childc.Name == Capacity)
                        {
                                capacity = decimal.Parse(childc.Text) * 1000;
                        }

                        if (childc.Name == DENSITY)


                            {

                                if (childc.Text != "" && childc.Text != "0")
                                {
                                    dty = decimal.Parse(childc.Text);

                            }


                            }
                         if (childc.Name == Quantity)

                            {


                            qtydecimal = decimal.Parse(childc.Text);

                            }

                         if (childc.Name == Mass)
                            {


                            mass = dty * qtydecimal;
                            childc.Text = mass.ToString();
                            dty = 0;

                        }
```

```
                    if(childc.Name == Haul)
                        {

                            dist =   fillDistance(c.Text );
                            childc.Text = dist.ToString();

                        }

                   if (childc.Name == Trip && capacity !=0 && mass!=0)
                   {
                       trips = Convert.ToInt32(mass/capacity);
                       childc.Text = trips.ToString();
                       qtydecimal = 0;
                       mass = 0;





                   }


               }
               }
           }
       }
```

6.11.1.15 Collect all Materials from Revit (Material Names)

FUNCTION USED LATER (COLLECT ALL MATERIALS)

```
private void collectallmaterials(string materialname)
        {
            if(countm == 0)
            {
                countm++;
                arrMaterial.Add(materialname);
                AllMaterialName =  materialname;
            }
            else
            {
                bool flag = false;
                foreach (var obj in arrMaterial.ToArray())
                {

                    string mobj = obj.ToString();
                    if (mobj == materialname)
                    {
                        flag = true;
                    }

                }

                if(flag == false)
                {
                    arrMaterial.Add(materialname);
                    AllMaterialName = AllMaterialName + "," + materialname;
                }
```

```
            }

        }
```

FUNCTION TO COLLECT REVIT DATA FOR WALLS

```
private void RevitdataWalls()
        {
            try
            {
                DataTable dt = new DataTable();
                dt.Columns.Add("Id", typeof(string));
                dt.Columns.Add("Material", typeof(string));
                dt.Columns.Add("SAId", typeof(string));

                dtAllwalls.Columns.Add("Id", typeof(string));
                dtAllwalls.Columns.Add("Material", typeof(string));
                dtAllwalls.Columns.Add("SAId", typeof(string));
                dtAllwalls.Columns.Add("ElementId", typeof(string));

                IList<Element> m_elementsToProcess;
                IList<Material> m_materialsToProcess;

                FilteredElementCollector        collector       =       new
FilteredElementCollector(m_doc);
                m_elementsToProcess                                       =
collector.OfCategory(BuiltInCategory.OST_Walls).WhereElementIsNotElementTyp
e().ToElements();

                int i = m_elementsToProcess.Count;
                MessageBox.Show(i.ToString() + " walls ");

                foreach (Element e in m_elementsToProcess)
                {

                    ElementId elementId = e.Id;


                    ICollection<ElementId>            materials            =
e.GetMaterialIds(false);

                    foreach (ElementId materialId in materials)
                    {

                        Material material = m_doc.GetElement(materialId) as
Material;

                        string name = material.Name.Replace(',', ':');
                        string SAId = "";
                        string strength = "";
                        SAId = material.StructuralAssetId.ToString();
                        if (SAId != null && SAId != "-1")
                        {
                            PropertySetElement           property           =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                            Parameter       paramConcreteCompression       =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);
```

```
                            strength                                       =
paramConcreteCompression.AsValueString();
                        }

                        int? mid = int.Parse(materialId.ToString());

                        bool flag = false;

                        foreach (DataRow table in dt.Rows)
                        {
                            int? dtmid = int.Parse(table[0].ToString());
                            if (mid == dtmid)
                            {
                                flag = true;
                            }

                        }

                        if (flag == false)
                        {


                            dt.Rows.Add(materialId.ToString(),         name,
strength);

                            collectallmaterials(name);


                        }
                        dtAllwalls.Rows.Add(materialId.ToString(),     name,
strength, e.Id.ToString());

                    }

                }




                comboBox2.DataSource = dt;
                comboBox2.ValueMember = "Id";
                comboBox2.DisplayMember = "Material";



            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }

        }
```

FUNCTION TO CALCULATE REVIT DATA FOR FLOORS

```
   private void RevitdataFloors()
        {
            try
            {
```

```
                DataTable dt = new DataTable();
                dt.Columns.Add("Id", typeof(string));
                dt.Columns.Add("Material", typeof(string));
                dt.Columns.Add("SAId", typeof(string));

                dtAllfloors.Columns.Add("Id", typeof(string));
                dtAllfloors.Columns.Add("Material", typeof(string));
                dtAllfloors.Columns.Add("SAId", typeof(string));
                dtAllfloors.Columns.Add("ElementId", typeof(string));

                IList<Element> m_elementsToProcess;
                IList<Material> m_materialsToProcess;

                FilteredElementCollector        collector        =        new
FilteredElementCollector(m_doc);
                m_elementsToProcess                                        =
collector.OfCategory(BuiltInCategory.OST_Floors).WhereElementIsNotElementTy
pe().ToElements();

                int i = m_elementsToProcess.Count;
                MessageBox.Show(i.ToString() + " floors ");
                foreach (Element e in m_elementsToProcess)
                {

                    ElementId elementId = e.Id;



                    ICollection<ElementId>            materials              =
e.GetMaterialIds(false);

                    foreach (ElementId materialId in materials)
                    {

                        Material material = m_doc.GetElement(materialId)  as
Material;
                        string name = material.Name.Replace(',', ':');
                        string SAId = "";
                        string strength = "";
                        SAId = material.StructuralAssetId.ToString();
                        if (SAId != null && SAId != "-1")
                        {
                            PropertySetElement           property          =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                            Parameter       paramConcreteCompression       =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);

                            strength                                       =
paramConcreteCompression.AsValueString();
                        }

                        int? mid = int.Parse(materialId.ToString());

                        bool flag = false;

                        foreach (DataRow table in dt.Rows)
                        {
                            int? dtmid = int.Parse(table[0].ToString());
                            if (mid == dtmid)
```

```
                            {
                                flag = true;
                            }

                        }

                        if (flag == false)
                        {


                            dt.Rows.Add(materialId.ToString(),         name,
strength);

                            collectallmaterials(name);
                        }
                        dtAllfloors.Rows.Add(materialId.ToString(),    name,
strength, e.Id.ToString());

                    }

                }



                ddlMaterialFloor.DataSource = dt;
                ddlMaterialFloor.ValueMember = "Id";
                ddlMaterialFloor.DisplayMember = "Material";

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }

        }
```

FUNCTION TO CALCULATE REVIT DATA FOR ROOF

```
        private void RevitdataRoofs()
        {
            try
            {
                DataTable dt = new DataTable();
                dt.Columns.Add("Id", typeof(string));
                dt.Columns.Add("Material", typeof(string));
                dt.Columns.Add("SAId", typeof(string));

                dtAllroofs.Columns.Add("Id", typeof(string));
                dtAllroofs.Columns.Add("Material", typeof(string));
                dtAllroofs.Columns.Add("SAId", typeof(string));
                dtAllroofs.Columns.Add("ElementId", typeof(string));

                IList<Element> m_elementsToProcess;
                IList<Material> m_materialsToProcess;

                FilteredElementCollector       collector       =       new
FilteredElementCollector(m_doc);
                m_elementsToProcess                                      =
collector.OfCategory(BuiltInCategory.OST_Roofs).WhereElementIsNotElementTyp
e().ToElements();

                int i = m_elementsToProcess.Count;
                MessageBox.Show(i.ToString() + " roofs ");
```

```
                foreach (Element e in m_elementsToProcess)
                {
                    ElementId elementId = e.Id;


                    ICollection<ElementId>            materials              =
e.GetMaterialIds(false);

                    foreach (ElementId materialId in materials)
                    {

                        Material material = m_doc.GetElement(materialId) as
Material;
                        string name = material.Name.Replace(',', ':');
                        string SAId = "";
                        string strength = "";
                        SAId = material.StructuralAssetId.ToString();
                        if (SAId != null && SAId != "-1")
                        {
                            PropertySetElement          property            =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                            Parameter       paramConcreteCompression        =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);

                            strength                                         =
paramConcreteCompression.AsValueString();
                        }

                        int? mid = int.Parse(materialId.ToString());

                        bool flag = false;

                        foreach (DataRow table in dt.Rows)
                        {
                            int? dtmid = int.Parse(table[0].ToString());
                            if (mid == dtmid)
                            {
                                flag = true;
                            }

                        }

                        if (flag == false)
                        {


                            dt.Rows.Add(materialId.ToString(),         name,
strength);

                            collectallmaterials(name);
                        }
                        dtAllroofs.Rows.Add(materialId.ToString(),     name,
strength, e.Id.ToString());

                    }

                }
```

```
                ddlMaterialRoof.DataSource = dt;
                ddlMaterialRoof.ValueMember = "Id";
                ddlMaterialRoof.DisplayMember = "Material";
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }

        }
```

FUNCTION TO CALCULATE REVIT DATA FOR COLUMNS

```
        private void RevitdataColumns()
        {
            try
            {
                DataTable dt = new DataTable();
                dt.Columns.Add("Id", typeof(string));
                dt.Columns.Add("Material", typeof(string));
                dt.Columns.Add("SAId", typeof(string));

                dtAllcolumns.Columns.Add("Id", typeof(string));
                dtAllcolumns.Columns.Add("Material", typeof(string));
                dtAllcolumns.Columns.Add("SAId", typeof(string));
                dtAllcolumns.Columns.Add("ElementId", typeof(string));

                IList<Element> m_elementsToProcess;
                IList<Material> m_materialsToProcess;

                FilteredElementCollector        collector        =        new
FilteredElementCollector(m_doc);
                m_elementsToProcess                                         =
collector.OfCategory(BuiltInCategory.OST_StructuralColumns).WhereElementIsN
otElementType().ToElements();

                int i = m_elementsToProcess.Count;
                MessageBox.Show(i.ToString() + " columns ");

                foreach (Element e in m_elementsToProcess)
                {

                    ElementId elementId = e.Id;


                    ICollection<ElementId>            materials               =
e.GetMaterialIds(false);

                    foreach (ElementId materialId in materials)
                    {

                        Material material = m_doc.GetElement(materialId) as
Material;
                        string name = material.Name.Replace(',', ':');
                        string SAId = "";
                        string strength = "";
                        SAId = material.StructuralAssetId.ToString();

                        if (SAId != null && SAId != "-1")
```

```
                        {
                            try
                            {
                                PropertySetElement        property        =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                                Parameter     paramConcreteCompression     =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);



                                strength                                  =
paramConcreteCompression.AsValueString();
                            }
                            catch
                            {


                            }



                        }



                        int? mid = int.Parse(materialId.ToString());

                        bool flag = false;


                        foreach (DataRow table in dt.Rows)
                        {
                            int? dtmid = int.Parse(table[0].ToString());
                            if (dtmid != null)
                            {
                                if (mid == dtmid)
                                {
                                    flag = true;
                                }
                            }


                        }

                        if (flag == false)
                        {



                            dt.Rows.Add(materialId.ToString(),         name,
strength);

                            collectallmaterials(name);
                        }

                        dtAllcolumns.Rows.Add(materialId.ToString(),   name,
strength, e.Id.ToString());

                    }
```

```
            }



            ddlMaterialColumn.DataSource = dt;
            ddlMaterialColumn.ValueMember = "Id";
            ddlMaterialColumn.DisplayMember = "Material";
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }

    }
```

FUNCTION TO CALCULATE REVIT DATA FOR BEAMS

```
    private void RevitdataBeams()
    {
        try
        {
            DataTable dt = new DataTable();
            dt.Columns.Add("Id", typeof(string));
            dt.Columns.Add("Material", typeof(string));
            dt.Columns.Add("SAId", typeof(string));

            dtAllbeams.Columns.Add("Id", typeof(string));
            dtAllbeams.Columns.Add("Material", typeof(string));
            dtAllbeams.Columns.Add("SAId", typeof(string));
            dtAllbeams.Columns.Add("ElementId", typeof(string));

            ElementClassFilter        beamClassFilter        =        new
ElementClassFilter(typeof(FamilyInstance));

            ElementCategoryFilter       beamTypeFilter       =       new
ElementCategoryFilter(BuiltInCategory.OST_StructuralFraming);

            LogicalAndFilter          beamFilter          =           new
LogicalAndFilter(beamClassFilter, beamTypeFilter);

            IEnumerable<Element> elems = from elem in
                                         new
FilteredElementCollector(m_doc).WherePasses(beamFilter).ToElements()

                                     select elem;

            int i = elems.Count();
            MessageBox.Show(i.ToString() + " beams ");

            foreach (Element e in elems)
            {

                ElementId elementId = e.Id;


                ICollection<ElementId>            materials             =
e.GetMaterialIds(false);
```

```
                    foreach (ElementId materialId in materials)
                    {

                        Material material = m_doc.GetElement(materialId) as
Material;
                        string name = material.Name.Replace(',', ':');
                        string SAId = "";
                        string strength = "";
                        SAId = material.StructuralAssetId.ToString();
                        if (SAId != null && SAId != "-1")
                        {
                            PropertySetElement           property           =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;

                            Parameter        paramConcreteCompression        =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);

                            strength                                        =
paramConcreteCompression.AsValueString();
                        }

                        int? mid = int.Parse(materialId.ToString());

                        bool flag = false;

                        foreach (DataRow table in dt.Rows)
                        {
                            int? dtmid = int.Parse(table[0].ToString());
                            if (mid == dtmid)
                            {
                                flag = true;
                            }

                        }

                        if (flag == false)
                        {


                            dt.Rows.Add(materialId.ToString(),         name,
strength);
                            collectallmaterials(name);
                        }
                        dtAllbeams.Rows.Add(materialId.ToString(),     name,
strength, e.Id.ToString());

                    }

                }


                int u = dt.Rows.Count;


                ddlMaterialBeam.DataSource = dt;
                ddlMaterialBeam.ValueMember = "Id";
                ddlMaterialBeam.DisplayMember = "Material";

            }
            catch (Exception ex)
            {
```

```
                MessageBox.Show(ex.ToString());
            }

        }
```

FUNCTION TO CALCULATE REVIT DATA FOR FOUNDATION

```
        private void RevitdataFoundation()
        {
            try
            {
                DataTable dt = new DataTable();
                dt.Columns.Add("Id", typeof(string));
                dt.Columns.Add("Material", typeof(string));
                dt.Columns.Add("SAId", typeof(string));

                dtAllfoundations.Columns.Add("Id", typeof(string));
                dtAllfoundations.Columns.Add("Material", typeof(string));
                dtAllfoundations.Columns.Add("SAId", typeof(string));
                dtAllfoundations.Columns.Add("ElementId", typeof(string));

                IList<Element> m_elementsToProcess;
                IList<Material> m_materialsToProcess;

                FilteredElementCollector        collector        =        new
FilteredElementCollector(m_doc);
                m_elementsToProcess                                          =
collector.OfCategory(BuiltInCategory.OST_StructuralFoundation).WhereElement
IsNotElementType().ToElements();

                int i = m_elementsToProcess.Count;
               MessageBox.Show(i.ToString() + " foundations ");

                foreach (Element e in m_elementsToProcess)
                {

                    ElementId elementId = e.Id;


                    ICollection<ElementId>            materials              =
e.GetMaterialIds(false);

                    foreach (ElementId materialId in materials)
                    {

                        Material material = m_doc.GetElement(materialId) as
Material;
                        string name = material.Name.Replace(',', ':');
                        string SAId = "";
                        string strength = "";
                        SAId = material.StructuralAssetId.ToString();

                        if (SAId != null && SAId != "-1")
                        {
                            try
                            {
                                PropertySetElement         property          =
m_doc.GetElement(material.StructuralAssetId) as PropertySetElement;
```

```
                                Parameter     paramConcreteCompression     =
property.get_Parameter(BuiltInParameter.PHY_MATERIAL_PARAM_CONCRETE_COMPRES
SION);


                                strength                                   =
paramConcreteCompression.AsValueString();
                            }
                            catch
                            {


                            }



                        }



                        int? mid = int.Parse(materialId.ToString());

                        bool flag = false;



                        foreach (DataRow table in dt.Rows)
                        {
                            int? dtmid = int.Parse(table[0].ToString());
                            if (dtmid != null)
                            {
                                if (mid == dtmid)
                                {
                                    flag = true;
                                }
                            }


                        }

                        if (flag == false)
                        {



                            dt.Rows.Add(materialId.ToString(),         name,
strength);

                            collectallmaterials(name);
                        }

                        dtAllfoundations.Rows.Add(materialId.ToString(),
name, strength, e.Id.ToString());

                    }

                }



                ddlMaterialFoundation.DataSource = dt;
```

```
                ddlMaterialFoundation.ValueMember = "Id";
                ddlMaterialFoundation.DisplayMember = "Material";
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }

        }
```

6.11.1.16 Function to Fill all fields for material energy tab except quantity

```
private void fillEEFields(string materialname, string ename)
        {

            string subtype = "";
            if (ename == "walls")
            {
                subtype = txtSubType.Text;
            }
            else if (ename == "floors")
            {
                subtype = txtSubTypeFloor.Text;
            }
            else if (ename == "roofs")
            {
                subtype = txtSubTypeRoof.Text;
            }
            else if (ename == "columns")
            {
                subtype = txtSubTypeColumn.Text;
            }
            else if (ename == "beams")
            {
                subtype = txtSubTypeBeam.Text;
            }
            else if (ename == "foundations")
            {
                subtype = txtSubTypeFoundation.Text;
            }
            if (subtype != "")
            {

                Char[]    strarr    =    subtype.ToCharArray().Where(c    =>
Char.IsDigit(c) || Char.IsPunctuation(c)).ToArray();
                decimal number = Convert.ToDecimal(new string(strarr));


                string name = "";

                SqlDataAdapter adapter = new SqlDataAdapter();
                DataSet ds = new DataSet();
                string sql = null;
                sql = "  SELECT TOP 1 * FROM EmbEnergy d CROSS APPLY(SELECT
CAST(replace(d.SubType, 'C', '') AS decimal) as Strength) a WHERE Material =
'Concrete' and EnglishName like '%" + name + "%' ORDER BY ABS(a.Strength - "
+ number + ") ";
                connection = new SqlConnection(connetionString);
```

```
                    connection.Open();
                    command = new SqlCommand(sql, connection);
                    adapter.SelectCommand = command;
                    adapter.Fill(ds);
                    adapter.Dispose();
                    command.Dispose();
                    connection.Close();


                    foreach (DataRow table in ds.Tables[0].Rows)
                    {

                        if (ename == "walls")
                        {
                            txtEECoefficient.Text                              =
table["EECoefficient"].ToString();
                            txtDensity.Text = table["Density"].ToString();
                            txtEName.Text = table["EnglishName"].ToString();
                            txtCName.Text = table["ChineseName"].ToString();
                            txtCode.Text = table["Code"].ToString();
                        }
                        else if (ename == "floors")
                        {
                            txtEECoefficientFloor.Text                         =
table["EECoefficient"].ToString();
                            txtDensityFloor.Text = table["Density"].ToString();
                            txtENameFloor.Text = table["EnglishName"].ToString();
                            txtCNameFloor.Text = table["ChineseName"].ToString();
                            txtCodeFloor.Text = table["Code"].ToString();
                        }
                        else if (ename == "roofs")
                        {
                            txtEECoefficientRoof.Text                          =
table["EECoefficient"].ToString();
                            txtDensityRoof.Text = table["Density"].ToString();
                            txtENameRoof.Text = table["EnglishName"].ToString();
                            txtCNameRoof.Text = table["ChineseName"].ToString();
                            txtCodeRoof.Text = table["Code"].ToString();
                        }
                        else if (ename == "columns")
                        {
                            txtEECoefficientColumn.Text                        =
table["EECoefficient"].ToString();
                            txtDensityColumn.Text = table["Density"].ToString();
                            txtENameColumn.Text                                =
table["EnglishName"].ToString();
                            txtCNameColumn.Text                                =
table["ChineseName"].ToString();
                            txtCodeColumn.Text = table["Code"].ToString();
                        }
                        else if (ename == "beams")
                        {
                            txtEECoefficientBeam.Text                          =
table["EECoefficient"].ToString();
                            txtDensityBeam.Text = table["Density"].ToString();
                            txtENameBeam.Text = table["EnglishName"].ToString();
                            txtCNameBeam.Text = table["ChineseName"].ToString();
                            txtCodeBeam.Text = table["Code"].ToString();
                        }
                        else if (ename == "foundations")
                        {
```

```
                        txtEECoefficientFoundation.Text                    =
table["EECoefficient"].ToString();
                        txtDensityFoundation.Text                          =
table["Density"].ToString();
                        txtENameFoundation.Text                            =
table["EnglishName"].ToString();
                        txtCNameFoundation.Text                            =
table["ChineseName"].ToString();
                        txtCodeFoundation.Text = table["Code"].ToString();
                    }

                }

            }

        }
```

APPENDIX B : SURVEY TOOL

Performa for Data Collection of Excavation Activity

<table>
<tr><th colspan="3">Data Collection for Excavation</th></tr>
<tr><td>Project/Location</td><td>Temperature</td><td>Humidity</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Daily Work Time</td><td>Soil Type</td><td>No. of Teams</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Depth</td><td>Dilation Co.</td><td>Total Excavation Volume</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Team Productivity</td><td>Excavator per team</td><td>Labor per team</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Distance from site to dump</td><td>Preparation T need before Excavation</td><td>Normal trucks need per day</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><th colspan="3">Excavation Zones</th></tr>
<tr><td rowspan="4">Zone ___</td><td>Area m^2</td><td></td></tr>
<tr><td>Volume of soil m^3</td><td></td></tr>
<tr><td>Start Date</td><td></td></tr>
<tr><td>Finish Date</td><td></td></tr>
</table>

Location of major material suppliers/manufacturers

Material	Supplier/Manufacturer	Geo Location
Concrete		
Steel Reinforcement		
Aluminium		
Windows/Doors		
Formwork		
Scaffolding		
Construction Machinery Equip.		

Data Collection for Scaffoldings

Type of scaffold used in the Project: ____________________________

Scaffold Dimensions:

Diameter (mm): ____________________________

Wall Thickness (mm): ____________________________

Mass per unit length (Kg/m): ____________________________

Data Collection for Formwork

Element	**Formwork Contact Area (m^2/m^3)**
Column	_____
Beam	_____
Slab	_____
Foundation	_____
Wall	_____