A CAD-BASED SOFTWARE FRAMEWORK FOR ESTIMATING ENERGY THROUGH A PRODUCT LIFE CYCLE

By

HE HUANG

A dissertation submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Mechanical and Materials Engineering

MAY 2012
To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of HE HUANG find it satisfactory and recommend that it be accepted.

__________________________
Gaurav Ameta, Ph.D., Chair

__________________________
Sankar Jayaram, Ph.D.

__________________________
Uma Jayaram, Ph.D.

__________________________
Ananth Kalyanaraman, Ph.D.
ACKNOWLEDGMENTS

I would like to express my gratitude to my thesis advisor Professor Gaurav Ameta for his guidance and encouragement during my PhD study. Without his help, I would not have been able to learn the intricacies of CAD based software development.

I would also like to thank the members of my thesis committee, Professor Sankar Jayaram, Professor Uma Jayaram, and Professor Ananth Kalyanaraman, for sharing their expertise. Their comments and suggestions helped to keep my research on track and focused.

I would like to thank my other lab mates in the Sustainable System Design Laboratory, and all of my friends, teachers, colleagues and Dr. James Chang for their companionship and for making life a worthwhile and memorable one. Finally I would like to thank my parents, and my fiancée Wenjing, thank you for all your love and encouragement throughout the years. To my mother and grandmother in heaven, you taught me to keep faith while facing difficulties, I miss you so much, and your love and unlimited support made all this possible.
Abstract

Excessive energy consumption has become a worldwide issue in today’s design and manufacturing industry. A framework for estimating energy consumption that could later be used to integrate with CAD/CAM systems is in demand.

This research develops an energy estimation framework which can be attached with various energy computational tools for calculating energy consumptions during a product life-cycle, i.e., from extraction of raw materials, design and manufacturing, use, to recycling or disposal phases. The study first reviews literature and background in related areas. Then the approach of developing the framework is illustrated following software development life cycle (SDLC) steps. The framework aims to be domain independent and flexible so that it will be expandable for different manufacturing domains and customizable for users. Details of the framework and its computational tools in each phase are presented afterwards. Interaction between framework and its computational tools are also discussed. At last, test cases of industrial products are conducted to test
the validity and evaluate the performance of the framework. The total energy consumption during their life cycle will be calculated. Such estimation can be used to re-design the part and assemblies for energy efficiency.

With help of this framework, knowledge engineers who exert to integrate knowledge into computer systems can interpret domain-specific knowledge and share their expertise to improve the framework. The framework also assists users who have little knowledge about energy computations to estimate energy consumptions during the design stage. Energy estimations in each phase and during whole product life cycle can be made and energy efficiency of products can be improved by utilizing this framework.
# TABLE OF CONTENTS

ACKNOWLEDGMENTS ................................................................. iii
Abstract........................................................................................ iv

## TABLE OF CONTENTS .......................................................... vi

## LIST OF FIGURES ................................................................... xi

## LIST OF TABLES ..................................................................... xvii

## CHAPTER ONE: INTRODUCTION ............................................. 1

1.1 Sustainable Manufacturing ............................................... 1
1.2 Product Life Cycle (PLC) and Product Lifecycle Management (PLM) ................................................................. 2
1.3 Life Cycle Assessment (LCA) .............................................. 4
1.4 Energy Efficiency and Energy Implication ............................ 6
1.5 Current Modeling Tools and Frameworks in Sustainable Manufacturing ................................................................. 7
1.6 Software Development Life Cycle (SDLC) ........................... 10
1.7 Knowledge Management and XML ................................... 11
1.8 Problem Statement ............................................................. 12
1.9 Outline of Dissertation ...................................................... 13

## CHAPTER TWO: APPROACH OF DEVELOPING ENERGY ESTIMATION FRAMEWORK FOLLOWING SDLC STEPS ............................................. 15

2.1 Project Planning ................................................................. 15
2.1.1 Need for Energy Estimation Framework .......................... 16
2.1.2 Significance of Energy Estimation Framework ............... 17
2.1.3 Expected Outcomes ................................................................. 18
2.2 Requirements Analysis ................................................................. 18
  2.2.1 System Boundaries of the Framework ............................................. 19
  2.2.2 Structural Aim of Energy Estimation Framework ......................... 20
  2.2.3 Functional Aim of Energy Estimation Framework ......................... 22
2.3 Design and Development ............................................................... 23
  2.3.1 Framework Construction .......................................................... 23
  2.3.2 Characterization of Energy Estimation in Four Phases ...................... 25
2.4 Integration and Test ........................................................................ 30

CHAPTER THREE: ENERGY ESTIMATION FRAMEWORK ......................... 33
  3.1 Overall Design of Energy Estimation Framework .............................. 33
  3.2 Knowledge Management through XML ............................................ 36
    3.2.1 XML and XML Schema (XSD) .................................................. 36
    3.2.2 Knowledge Management Class: CXML .................................... 38
    3.2.3 Validation of XML ................................................................. 39
    3.2.4 Parsing of XML ........................................................................ 40
  3.3 User Interfaces of Energy Estimation Framework .............................. 41
  3.4 Knowledge Management Module of Energy Estimation Framework .... 47
  3.5 Energy Estimation Module of Energy Estimation Framework ............ 51

CHAPTER FOUR: COMPUTATIONAL TOOLS OF ENERGY ESTIMATION FRAMEWORK ................................................................................. 56
  4.1 Materials Extraction Phase Energy Estimation Tool ......................... 57
4.2 Design Phase Energy Estimation Tools

4.2.1 Overview of Energy Estimation in Design and Manufacturing Phase

4.2.2 Design Energy Estimation for Milling Operation

4.2.2.1 Steps to Compute Energy Estimates for Milling Operation

4.2.2.2 Algorithm for Calculating the Minimum Oriented Bounding Box

4.2.2.3 Case Studies for Energy Estimation in Design Phase for Milling Operation

4.2.3 Design Energy Estimation for Turning Operation

4.2.3.1 Steps to Compute Energy Estimates for Turning Operation

4.2.3.2 Algorithm for Calculating the Removal Volume

4.2.3.3 Case Studies for Energy Estimation in Design Phase for Turning Operation

4.2.4 Design Energy Estimation for Casting Operation

4.3 Manufacturing Phase Energy Estimation Tool Based on Manufacturing Process Plan

4.3.1 Manufacturing Process Planning

4.3.2 STEP-NC Process Plan Structure, XML and Its Schema
4.3.3 Methods for Calculating Manufacturing Energy in Machining Operation ........................................................... 95

4.3.3.1 Machining Energy Calculation ........................................ 95

4.3.3.2 Steps to Calculate Manufacturing Energy in Machining Operation ....................................................................... 97

4.3.4 Case Study ........................................................................... 98

4.4 Use Phase Energy Estimation Tool ........................................ 100

4.5 Disposal Phase Energy Estimation Tool .................................... 102

4.6 Product Life Cycle Energy Estimation Tool and Computational Results Feedback ................................................................. 103

CHAPTER FIVE: INTERACTION BETWEEN ENERGY ESTIMATION FRAMEWORK AND ENERGY COMPUTATIONAL TOOLS ..................... 105

5.1 Data Flow of Energy Estimation Framework ............................... 105

5.2 Communication Between Energy Estimation Framework and Its Computational Tools .......................................................... 107

CHAPTER SIX: INDUSTRIAL TEST CASES OF ENERGY ESTIMATION FRAMEWORK ...................................................................................... 110

6.1 Industrial Test Case 1: Cardan Adapter ...................................... 110

6.2 Industrial Test Case 2: Clamping Ring ....................................... 124

6.3 Industrial Test Case 3: Sliding Mount Units-Assembly .......... 128

CHAPTER SEVEN: CONCLUSIONS, DISCUSSIONS, AND FUTURE WORK ............................................................................................................. 133

7.1 Conclusions and Research Contributions ...................................... 133
7.2 Research Limitations ............................................................... 135
7.3 Recommendations for Future Work ............................................ 137
REFERENCES .................................................................................. 138
APPENDIX ........................................................................................ 143

Appendix A: Sample STEP-NC XML File for Process Plan ............ 143
LIST OF FIGURES

Figure 1: Four Phases of a Product Life Cycle .......................................................... 3
Figure 2: General SDLC Steps [41] ...................................................................... 10
Figure 3: Schematic Overview of Energy Estimation Framework ....................... 16
Figure 4: Generic Shell of Energy Estimation Framework ................................. 25
Figure 5: Use Case Scenarios of Energy Estimation Framework ....................... 29
Figure 6: Integration among Framework, Computational Tools, Knowledge Engineer, and Estimate User ................................................................. 30
Figure 7: Class Diagram of Energy Estimation Framework ............................... 34
Figure 8: Example of an XML File and its Schema ........................................... 37
Figure 9: Class Diagram of CXML Class ............................................................ 39
Figure 10: Algorithm of XML Parsing .................................................................. 41
Figure 11: Structure of Interfaces of the Energy Estimation Framework .......... 42
Figure 12: Scenario 1: Open Energy Estimation Framework ............................ 43
Figure 13: Interface of the Energy Estimation Framework ................................. 44
Figure 14: Scenario 2: Click New or Open a Part File ....................................... 44
Figure 15: Scenario 3: Client Type Selection ...................................................... 45
Figure 16: Scenario 3 (a): Client Type Selected as Energy Estimation User ...... 46
Figure 17: Scenario 3 (b): Client Type Selected as Knowledge Engineer .......... 47
Figure 18: Menu and Toolbar of Knowledge Management Module .................... 48
Figure 19: Functions in Knowledge Management Module .................................. 48
Figure 20: Scenario of Configuring XML File in Knowledge Management Module: Open ........................................................................................................... 49
Figure 21: Scenario of Configuring XML File in Knowledge Management Module: Edit and Save ............................................................................................................................... 50
Figure 22: Scenario of Using Knowledge Management Module: Load XML ...... 51
Figure 23: Menu and Toolbar for Energy Estimation Module ..................................... 52
Figure 24: Components in Energy Estimation Module ............................................. 53
Figure 25: Scenario of Using Energy Estimation Module: User Input Option ...... 54
Figure 26: Scenario of Using Energy Estimation Module: Computational Result 55
Figure 27: List of Tools that are attached to Energy Estimation Framework ...... 56
Figure 28: Energy Estimation Tool: Materials Extraction Phase ......................... 58
Figure 29: Methods to Calculate Extraction Phase Energy .................................. 59
Figure 30: Computational Tool Selection ............................................................. 60
Figure 31: Energy Estimation Tool: Design and Manufacturing Phase ............. 61
Figure 32: Range of Estimated Energy in Design and Manufacturing Phase .... 62
Figure 33: Overall Algorithm for Computing Energy and CW for Part and Assembly in Design Phase for Milling Operation ................................................ 64
Figure 34: Algorithm to Compute the Minimum Oriented Bounding Box ......... 66
Figure 35 (a) Initial Revolved Part as an Input to the Minimum Oriented Bounding Box Algorithm, (b) XZ Cut View of the Overlapping Minimum Oriented Bounding Box and the Revolved Part. .......................................................................................... 67
Figure 36: (a) Initial Part as an Input to the Minimum Oriented Bounding Box Algorithm, (b) The Convex Hull of the Part Computed Using Qhull ................. 67
Figure 37: (a) YZ Cut View of Bounding Box Using a Minimum Zone that Has the Minimum Distance for the 2D Projected Vertices (b) YZ Cut View of the Bounding
Box Using the Minimum Zone Obtained by Minimizing the Product of the Distances of All Possible Minimum Zones for the 2D Projected Vertices...........68
Figure 38: (a) Convex Hull for the Cover Plate (b) Fit of the Minimum Oriented Bounding Box and the Cover Plate. .................................................................................69
Figure 39: Effect of Fillet in Removal Volume for Convex and Concave Edges. 71
Figure 40: Assembly Case: Slider-crank Mechanism...........................................71
Figure 41: Overall Algorithm for Computing Energy and CW for Part and Assembly in Design Phase for Turning Operation.............................................................76
Figure 42: Algorithm to Compute the Removal Volume.........................................78
Figure 43: Example of Turnable Faces: (a) Cylindrical Face; (b) Conical Face; (c) Planar Perpendicular to Rotation Axis; .................................................................79
Figure 44: Circumferential Projections of Various Entities [72].............................80
Figure 45: (a) Part Containing Unturnable Faces; (b) Vertical Projection; (c) Circumferential Projection; (d) Revolved Minimum Turnable Entity .................81
Figure 46: (a) Turnable Part; (b) Wireframe Showing the Overlap of the Revolved Part and its Minimum Oriented Bounding Box; (c) Wireframe Showing the Overlap of the Revolved Part and its Minimum Bounding Cylinder; (d) Cut View Showing the Overlap of the Revolved Part and its Minimum Bounding Cylinder 82
Figure 47: (a) Solid and (b) Wireframe View of the Part at the Preliminary Design Stage. (c) Solid and (d) Wireframe View of the Part at the Detail Design Stage. 83
Figure 48: (a) Solid View of Minimum Turnable Entity; (b) Wireframe Showing the Overlap of Part and its Minimum Turnable Entity; (c) Wireframe Showing the
Figure 67: Input/Edit Tool Locations ................................................................. 113
Figure 68: Parse XML File ................................................................................ 114
Figure 69: (a) Switch Client Type and Input User Specified Data (b) User Input Data Stored by the Framework ................................................................. 115
Figure 70: (a) Extraction Phase Energy Estimation for Cardan Adapter (b) Result Stored by Extraction Phase Energy Estimation Tool ......................................... 116
Figure 71: Select Computational Tool ............................................................... 117
Figure 72: (a) Design and Manufacturing Phase Energy Estimation for Cardan Adapter (b) Result Stored by Design and Manufacturing Phase Energy Estimation Tool ................................................................. 119
Figure 73: (a) Use Phase Energy Estimation for Cardan Adapter (b) Result Stored by Use Phase Energy Estimation Tool .................................................. 120
Figure 74: (a) Disposal Phase Energy Estimation for Cardan Adapter (b) Result Stored by Disposal Phase Energy Estimation Tool ........................................... 121
Figure 75: PLC Energy Estimation for Cardan Adapter .................................... 122
Figure 76: PLC Computational Results for Cardan Adapter Stored by the Framework ........................................................................................................ 123
Figure 77: Computational Results Feedback for Cardan Adapter ..................... 124
Figure 78: Industrial Case: Clamping Ring ....................................................... 125
Figure 79: Computational Results in Design and Manufacturing Phase for Clamping Ring .................................................................................................. 126
Figure 80: Computational Results for Clamping Ring in Each Phase ............... 126
Figure 81: PLC Energy Distribution for Clamping Ring ..................................... 127
**LIST OF TABLES**

Table 1: Review of Literature ................................................................................ 9
Table 2: Factors at Raw Material Extraction Phase ............................................26
Table 3: Factors at Design and Manufacturing Phase .......................................27
Table 4: Factors at Use Phase ...........................................................................28
Table 5: Factors at Disposal Phase ....................................................................29
Table 6: Comparison of the Calculations of Range of Energy and CW Estimates of the Part at the Preliminary and Detail Design Stages .......................... 70
Table 7: Comparison of the Calculations of Range of Energy and CW Estimates of the Part at the Preliminary and Detail Design Stages .......................... 74
Table 8: Comparison of the range of energy and CW estimates of the part at the preliminary and detail design stages ......................................................... 85
Table 9: Comparison of the calculations of range of energy and CW estimates of the assembly at the preliminary and detail design stages .............................. 88
Table 10: Average Value of Cutting Force Constant C for Common Materials [77] .................................................................................................................. 95
CHAPTER ONE: INTRODUCTION

1.1 Sustainable Manufacturing

It has been reported that “Americans represent merely 5 percent of the world’s population, yet they consume 40 percent of Earth’s nonrenewable resources and produce nearly 30 percent of its total waste…” [1]. As energy consumptions increase at a significant rate, industries begin to seek ways to use energy efficiently, and comply with government regulations related to environment and human health while enhancing the marketability of their products and services. Sustainable manufacturing has become a strategic trend in manufacturing industry. Sustainable manufacturing is defined by the Department of Commerce as “the creation of manufactured products that use processes that are non-polluting, conserve energy and natural resources, and are economically sound and safe for employees, communities, and consumers.” [2]. According to Reich et al. [3], sustainable manufacturing strategy implements the following three important components: (1) selection and application of appropriate modeling metrics for measuring manufacturing sustainability, (2) completion of comprehensive, transparent, and repeatable life-cycle assessment (LCA), and (3) adjustment/optimization of the system to minimize environmental impacts and cost.

The literature above emphasizes the idea of sustainability, and it gives inspirations to develop a software framework to optimize energy consumption; such framework will help manufacturing industries to achieve sustainable strategy so that environmental impacts and cost can be minimized. The thesis
will give in-depth discussions on such energy estimation framework in the following chapters. The energy estimation framework can be regarded as an abstract shell for energy calculations. It can be attached with various energy computational tools for diverse purposes. The framework defines rules for energy calculations and interactions between users and software so that it is reusable and expandable. Before probing into the details of the framework, several concepts need to be introduced in the first place. The following sections will present the concepts of Product Life Cycle (PLC), Product Lifecycle Management (PLM), and Life Cycle Assessment (LCA) first, followed by the literature review of energy efficiency and current modeling tools and frameworks in sustainable manufacturing which indicate the significance for developing such a software framework. Then, Software Development Life Cycle (SDLC) will be discussed to show how to turn the idea into the implementation of software. Implementation of the software requires knowledge management, and Extensible Markup Language (XML) is adopted as a knowledge communicator between users and software. Thus, the chapter will shed lights on XML and knowledge management at last.

1.2 Product Life Cycle (PLC) and Product Lifecycle Management (PLM)

In general, the Product Life Cycle (PLC) for manufacturing products includes four phases. The four phases are materials extraction phase, design and manufacturing phase, use phase, and disposal or recycling phase [4]. These four phases can be connected through supply chain. Figure 1 illustrates the generic life cycle of a product.
The concept of product life cycle takes into account all the relevant impacts on the environment, economy, and the society during the whole chain of a product’s life cycle [4]. It enables industry service providers, managers, and customers to make decisions for the longer term with considerations of comprehensive perspectives during a whole product chain. A more sustainable direction of product and process development can be brought so that industries can harvest the benefits of developing and applying cleaner process and product options.

Figure 1: Four Phases of a Product Life Cycle

With the concept of PLC, the idea of Product Lifecycle Management (PLM) has come up. PLM is a process to manage PLC. It has developed to manage the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal [5]. PLM integrates people, data and processes to provide comprehensive product information [6].
Documented benefits of product lifecycle management include [7]:

- Reduced time to market;
- Improved product quality;
- Reduced costs;
- Reusability of original data;
- A framework for product optimization;
- Reduced environmental impacts/wastes;

There are various applications of PLM in both research and business fields, such as product design (CAX), manufacturing process management, product data management and systems engineering. Current research shows that lots of modeling and analysis tools have been developed to support PLM. The following section will introduce some popular Life Cycle Assessment (LCA) tools applied in PLC.

1.3 Life Cycle Assessment (LCA)

According to Khan et al., [8] life cycle assessment (LCA) is one of the most important techniques for successful implementation of a process or product development in the context of environmental sustainability. The methodological framework for life cycle assessment is defined by the ISO 14040 series, which includes goal and scope definition, inventory analysis, impact assessment, and interpretation [9]. Rebebitzer et al. [10] reviewed the framework, goal and scope definition, inventory analysis and applications of life cycle assessment. The goal and scope definition of LCA provide system boundaries and functional units which define the exact contents that are being studied and quantities to be
delivered. Life cycle inventory analysis gives estimations of the consumptions of resources and the amount of waste/emissions which are attributable to a product's life cycle [10]. Life cycle impact assessment indicates the potential contributions of the resource extractions and wastes/emissions based on an inventory analysis [10]. Life cycle interpretation summarizes the results from life cycle inventory analysis and/or life cycle impact assessment and gives conclusions and recommendations for the LCA study [9]. Due to the importance of LCA during a product development process, LCA has been applied in various aspects such as product design and manufacturing. Bevilacqua et al. [11] developed a sustainable product life cycle model combining environmental and economic considerations in manufacturing firms. There are a lot of existing LCA software, like Sima Pro [12], Gabi [13], and EcoScan [14], allowing the Life-Cycle assessments at high level of details. Many research projects integrate LCA with CAD for users to apply decisions at design stage. Nawata and Aoyama [15] developed a Life-Cycle design system for machine tools by linking LCA data to CAD/CAM data. Leibrecht [16] discussed basic principles for integrating life cycle assessments into CAD-based product model.

Despite the large application of LCA, it has its disadvantages. A product designer cannot perform LCA alone without LCA experts, designers, manufacturers, suppliers and management. Moreover, LCA only provides methodology for assessing the environmental impacts and does not provide any tools or metrics for reducing the impacts [16].
1.4 Energy Efficiency and Energy Implication

The previous sections introduce basic concepts of PLC, PLM, and LCA. In order to implement the energy estimation software mentioned in section 1.1, current researches for improving energy efficiency need to be studied. The following two sections will review literature on energy efficiency and current modeling tools and frameworks in sustainable manufacturing.

Energy is expected to sustain every aspect of a product’s life cycle. A study conducted by the National Association of Manufacturers (NAM, www.nam.org) speculated possible savings of about 19 billion dollars with development of new energy efficient technologies [17]. Energy Star [18] standard has been developed in an attempt to reduce energy consumption. However, Energy Star does not include the energy consumption during the manufacturing phase of a product life cycle.

Extensive research has been conducted to utilize energy efficiently during manufacturing processes. Munoz & Sheng [19] found an analytical approach to determine the environmental impact of machining; Sheng et al. [20] investigated an environmental-based systems planning for machining. However, the research focused on energy consumption only in the process of material removal. Chen et al. [21] created a new all life cycle energy-consuming model to estimate energy consumption during the whole product life-cycle. But the authors only focused on the theoretical portion and have not performed the energy estimation in the manufacturing phase. Gutowski’s research group investigated various energy impacts of different manufacturing processes and made an environmental
analysis of machining [22-25], they generalized the energy consumption of diverse manufacturing processes. Their research focuses mainly at manufacturing stage without concerning energy estimation in design stage and of course is not specific to products. Jeswiet [26] generalized the computation of CO₂ factors from electricity requirements for product related CO₂ emissions. However, the author did not consider the variability of manufacturing processes.

1.5 Current Modeling Tools and Frameworks in Sustainable Manufacturing

Currently, a large number of modeling tools are available and in use in sustainable manufacturing. For example, Liu et al. [27] created a generic model for analyzing consumption situation of product material resources in manufacturing system. Xu et al. [28] presented a feature-based life-cycle assessment model in practice. Based on this model, life-cycle assessment data can be generated from life-cycle inventory analysis, and then the model can feed the data back to manufacturing planning stages to improve the machining plan. Dimache et al. [29] developed a life cycle cost estimation tool for decision-making in early phases of design process. Feng et al. [30] modeled cost and process information for dry machining. Reddy et al. [31] modeled the domain knowledge for DfM of castings. Zhao and Shah [32] researched on modeling and representation of manufacturing knowledge for DfM systems to help the manufacturing knowledge engineers formulize their knowledge and store it into computer, and help the designers analyze manufacturability during design stage. However, the researchers above either concentrated on specific domain of
manufacturing or drew attention only to particular phase of a product life cycle. Therefore, this paper proposes to construct a generic and domain independent framework during the whole product life cycle. The details of the software framework for energy estimation will be illustrated in chapter two.

Numerous researchers have proposed diverse frameworks for sustainable manufacturing. Yang and Lin [33] developed an integrated framework for feature-based early manufacturing cost estimation. This system tends to provide a tool to assist designers to estimate the cost during design stage in order to downstream the unnecessary costs. Shin and Waters [34] developed a framework of a machining advisory system which could combine rules for inferring optimal machining parameters. Lee et al. [35] presented a computational framework that supports concurrent mold manufacturing process planning. This research will facilitate the automation of the mold development process planning, thus improving the efficiency and quality of molding product. Wei and Egbelu [36] proposed a framework for estimating manufacturing cost from geometric design data. Kwon et al. [37] developed a grammar-based framework for integrating design and manufacturing. It helps engineers to navigate through the process alternatives and to foresee the effects of different design choices. Tharakan et al. [38] proposed a manufacturability evaluation shell which provides a reconfigurable environment for both technical and economic manufacturability evaluation. Zhao et al. [39] developed a normative DfM framework based on benefit-cost analysis to deal with the trade-offs between design objectives and manufacturing cost/efficiency. Yang and Lee [40] proposed a feature modification
framework for generation of alternative process plans. The achievements above show a significant research progress in sustainable manufacturing. However, none of these frameworks can be directly utilized for energy estimation. In order to estimate energy consumption, a computational software framework needs to be constructed; chapter two will present the approach for developing such energy estimation framework.

Below is a brief summary of the reviewed literature above:

Table 1: Review of Literature

<table>
<thead>
<tr>
<th>Related Literatures</th>
<th>Conducted Researches</th>
<th>Examples and References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sustainable Manufacturing</td>
<td>Americans consume 40% of earth’s energy; Three components: Modeling Metrics, LCA, Environmental Impacts and Cost;</td>
<td>Orr, 2007 Reich et al., 2008</td>
</tr>
<tr>
<td>Current Modeling Tools in Sustainable Manufacturing</td>
<td>Material resources consumption; Life-cycle cost and process information; Domain knowledge of DfM for specific manufacturing operations;</td>
<td>Liu, 2002 Dimache, 2007; Feng, 2000 Zhao, 2004: Machining; Reddy, 2007: Casting</td>
</tr>
<tr>
<td>Life Cycle Analysis (LCA)</td>
<td>Commercial use; Principle for integrating LCA into CAD; Linking of LCA data to CAD/CAM data; Drawbacks: system boundaries, data issue, large involvements, does not reduce impacts</td>
<td>Simapro, EcologiCAD Leibrecht, 2005 Nawata and Aoyama Mayer, 2008,</td>
</tr>
</tbody>
</table>
1.6 Software Development Life Cycle (SDLC)

Since the research objective is to construct a software framework to estimate energy consumptions and the software framework will be developed using SDLC steps, this section will give a brief overview of SDLC; the detailed discussion on how to implement the software using SDLC will be presented in chapter 2.

SDLC is a complete plan outlining how the software will be created, developed, implemented and eventually retired from its function [41]. Figure 2 illustrates general SDLC steps. In project planning stage, the application goal will be defined, and feasibility will be analyzed. Requirements definition stage characterizes the need, requirements, and boundary conditions for the project. In design and development stage, software model, class diagram and use cases are proposed, and then codes are implemented. Integration and test stage
ensures the smooth interaction between different modules of software and validates the software through various test cases. Installation and acceptance stage is responsible for distributing and deploying the software.

1.7 Knowledge Management and XML

In order to make energy estimations, the software framework needs to deal with a lot of input from existing knowledge. Therefore this framework requires Knowledge Management (KM). XML (eXtensible Markup Language) is adopted as the communicator to interpret and transfer knowledge. This section will give a brief introduction on KM and XML.

Knowledge Management is getting the right information to the right people at the right time, and helping people create knowledge and share and act upon information in ways that will measurably improve the performance [42].

KM activities can identify and capture the information that exists across the knowledge domain [42]. KM processes can help people find, organize, and share the knowledge resources across the barriers of time and space.

XML (eXtensible Markup Language) is a specification developed by the World Wide Web Consortium (W3C) [43]. XML allows users to define their own descriptive tags. With XML, it is possible to transport and store data [44]. The XML specification defines an XML document as a well-formed text, which satisfies a predefined syntax rule [45]. The predefined rules are provided in the specification called XML Schema [45]. XML Schema is one of the validation methods for XML file. It acts like a grammar for XML file. It uses a rich data
typing system and allows for detailed constraints on an XML document's logical structure [46].

XML is adopted to store and transform knowledge for the software because it is able to handle information that needs to be read in by software. Self-defined tags and structures can be created so that modifying data files will become convenient and flexible. For example, an XML structure of a part data can be defined as:

```xml
<Part>
  <Material Class…>
    <Material Name>…</Material Name>
    ...
  </Material Class…>
</Part>
```

The tags can be easily modified and expanded, by which knowledge can be successfully stored and transformed.

### 1.8 Problem Statement

According to the discussions above, the problem statement for this research can be described as below.

This research is aiming to develop a software based energy estimation framework which can be attached with various energy computational tools for calculating energy consumptions during a product life-cycle, i.e., from extraction of raw materials, design and manufacturing, use, to recycling or disposal phases. This software framework will be developed following software development life
cycle (SDLC) steps. The framework aims to be domain independent so that it will be expandable for different manufacturing domains. It will also be customizable for users. The total energy consumption during products' life cycle can be calculated. Such estimation can be used to re-design the part and assemblies for manufacturing energy efficiency.

1.9 Outline of Dissertation

The dissertation contains seven chapters. Each chapter illustrates a specific topic towards the completion of the energy estimation framework. Chapter 1 reviews literature and background in sustainable manufacturing, PLC, PLM, LCA, energy efficiency, modeling tools and frameworks in sustainable manufacturing, SDLC and XML. In Chapter 2, the approach of developing the framework is illustrated following software development life cycle (SDLC) steps. It starts with project planning to analyze the need, significance, and outcomes of the energy estimation framework. System boundaries, structural and functional aims are then proposed in requirements analysis. Chapter 2 also presents how to design, develop, integrate and test the framework. Details of the energy estimation framework and its computational tools are presented in chapter 3 and chapter 4 respectively. Chapter 3 introduces the overall design of the framework first, and then elaborately illustrates its user interface, knowledge management module, and energy estimation module. Chapter 4 illustrates all the computational tools that are attached to the framework to compute energy in each phase and whole product life cycle. In chapter 5, the interaction between framework and its computational tools are discussed. In chapter 6, test cases of
industrial products are conducted to test the validity and evaluate the performance of the framework. At last, the dissertation draws conclusion of current research and proposes recommendations for future work.
CHAPTER TWO: APPROACH OF DEVELOPING ENERGY ESTIMATION FRAMEWORK FOLLOWING SDLC STEPS

This chapter presents the approach for developing energy estimation framework following SDLC steps, i.e., from project planning, requirement analysis, to design, development, integration and test. In project planning phase, need and significance for developing such framework is proposed, and expected outcomes are listed. Requirement analysis phase provides system boundaries and analyzes structural and functional aims that the framework needs to reach. In design and development phase, it describes how to construct the framework and how to characterize the energy estimation functionality. Integration and testing are presented at last.

2.1 Project Planning

Energy is indispensable for transformation of raw materials to final products. In today’s design and manufacturing industry, energy consumption is increasing at a fast rate [47]. Energy is obtained from limited resources such as gas and coal, and its price has been constantly increasing [48]. Moreover, excessive energy consumption brings lots of environmental issues such as air pollution and landfilling of hazardous materials, which may endanger human health [47]. Thus, it is vital to reduce energy consumption during a product’s life cycle. It has also been reported that 60-80% of the products manufactured by original equipment manufacturers are fabricated by outside suppliers [49] working in different domains. Therefore, knowledge from each domain is isolated and cannot be reused [50]. If a product’s energy consumption can be estimated
inside a domain independent framework for its whole life cycle, energy can be utilized efficiently through the life cycle. As Figure 3 illustrated, users can optimize their designs to achieve appropriate performances with reasonable energy consumptions. Knowledge engineers can share their expertise to improve the energy estimation framework.

![Figure 3: Schematic Overview of Energy Estimation Framework](image)

### 2.1.1 Need for Energy Estimation Framework

According to Gielen et al. [51], almost 33% of the world’s energy consumption and 36% of its carbon dioxide (CO₂) emissions are attributable to manufacturing industries. In 2004, the global primary energy supply was 469 exajoules, of which more than 147 exajoules are accounted for industry consumptions [51]. Excessive energy consumption may result in [51]: (1) increasing net greenhouse gas emissions, (2) increasing acid deposition and air pollution, and (3) increasing landfilling. Greenhouse gas emissions result in global warming, and cause global climate problems. Nitrogen oxides, sulfur dioxide and mercury compounds that are exhausted into the air can significantly reduce air quality and threaten human health. Landfilling also causes ground pollution. Excessive energy consumption creates an added cost of dealing with
the emissions and pollutions. Therefore, it increases the total life-cycle cost of products.

According to the energy consumption trend [52], energy consumption is increasing for the next decades. Growing global demand for products, services, and public infrastructure are increasing the overall energy consumptions. Research data show that industry’s use of energy has grown by 61% between 1971 and 2004 [53]. Hence, excessive energy consumption is a pressing issue in today’s design and manufacturing industry. The manufacturers have the responsibility to manufacture products energy-efficiently and also emit low greenhouse gases. In order to produce energy-efficient products, planning with respect to energy consumptions has to be taken into account during the design stages. Thus, estimation of energy consumption during a whole product life-cycle will be in great demand, thereby reduces hazardous environmental impacts.

2.1.2 Significance of Energy Estimation Framework

Constructing an energy estimation framework during a product life-cycle will have a significant impact on both users and knowledge engineers. From the user’s point of view, estimation of energy consumption conducted during design stage will provide different alternatives for making decisions. From the knowledge engineer’s point of view, the proposed framework provides the capability to input domain knowledge. When energy consumption for the entire life-cycle of a product is decreased, the impacts of greenhouse gas emissions, air pollutions will also decrease, thereby reducing the total life-cycle cost of the products. To the best of the author’s knowledge, few tools have been developed to estimate
energy consumptions during a whole product life-cycle. Therefore both designers and knowledge engineers can significantly benefit from the proposed framework. Furthermore, manufacturing industry can assess their current manufacturing plans and adapt the optimal methods to process products by consuming less energy at a low cost.

2.1.3 Expected Outcomes

The expected key contributions from this work can be described as follows:

(a) A design framework for energy estimation during a product life cycle will be constructed.

(b) The framework will be customizable for energy estimation users.

(c) The framework will be expandable for developers and knowledge engineers.

(d) The framework will be domain independent, so that it will not depend on the specific domain knowledge.

(e) The framework will contain various tools to estimate energy consumptions during four phases of a product life-cycle and for diverse manufacturing processes.

(f) Given a case product, energy consumed in each phase and during its life cycle will be generated.

2.2 Requirements Analysis

The objective of this research is to build a domain independent design framework to estimate energy consumption. The framework should be able to compute energy consumptions at different phases in a product’s life cycle. These
phases include raw material extraction phase, design and manufacturing phase, use phase, and recycling or disposal phase. Environmental impact such as carbon weight during the life cycle should also be obtained. The estimated energy consumption can be significantly helpful for industries to assess energy efficiency of their products and improve their processing methods. Consequently, total energy consumption and hazardous environmental impacts will be reduced.

2.2.1 System Boundaries of the Framework

The boundaries of the framework are defined as follows:

(a) The framework defines a set of classes and the model for energy estimation, the structure will allow various tools to estimate energy consumptions at different phases of a product life cycle.

(b) The framework estimates direct energy only. Direct energy means the energy consumed directly for producing a part. Since machine is also made of parts, the associated energy consumed in making the machine’s parts will not be considered.

(c) The energy consumption of specific manufacturing processes can be computed only after the domain knowledge is input. The results estimated are related with the completion of domain knowledge.

(d) Actual energy consumption can be calculated only after detailed process plans such as manufacturing location and manufacturing equipments are fixed. It is impossible to obtain all the manufacturing information in design stage. Therefore, range of uncertainty for energy estimation exists and refinement of the results is related with the detailed process plans.
(e) Various individual impacts such as different setups and manufacturing operations will affect estimated results. This study will provide generic estimates based on the domain knowledge in typical situations. Unless detailed information for individual case is provided, the framework cannot make specific energy estimations.

(f) Presently, the framework will not integrate with experimental data and will not consider tolerance during design and manufacturing; the integration and tolerance is left for future work. Thus, estimates provided by the framework may differ from those in actual manufacturing experiments.

(g) Energy consumption during a product life-cycle is also related with the supply chain including location of manufacturers and distributors, transportation methods, and so on. Supply chain will not be considered in this proposed framework.

(h) Energy consumption for disassembling will not be considered in the proposed framework. At the end of a product life cycle, only energy consumption at disposal phase will be calculated.

(i) In this research, data for materials, manufacturing resources, and manufacturing methods are far from complete. However, due to the domain-independent feature of the framework, such information can be expanded through diverse knowledge engineers.

2.2.2 Structural Aim of Energy Estimation Framework

The specific aims include both structural and functional. The structural aim is to construct a customizable and expandable framework for energy estimation.
The functional aim is to compute energy consumption at four phases during a product life-cycle.

The structural aims for the proposed framework will be specified as follows:

(a) Domain Independency

The framework for energy estimation should be independent to specific domains. Although different manufacturing domains have diverse knowledge in energy computations, the framework should characterize information in diverse domains in a generic manner. Separation of the domain independent framework from specific domain knowledge will be a key aspect for the framework to be generic.

(b) The framework should be expandable for developers and knowledge engineers

In order that the framework is generic, it must be designed so that different domain knowledge can be easily modified, updated and expanded. Each knowledge engineer has his specialty and should be able to expand knowledge in different domains. Developers should be able to implement various computational tools and attach the tools to the framework so that the framework can be expanded. The capability of modification on existing knowledge and addition of new domain knowledge and new computational tools is vital to make the framework achieve different objectives successfully.

(c) The framework should be customizable for users

The framework should offer an environment where users can configure and customize according to their needs so that the output is personalized.
(d) Flexibility

The framework should be able to make energy estimations at multiple stages such as preliminary design stage and detailed design stage. If a designer wants to modify some of his design methods or do a redesign, the system should provide real-time feedbacks on any changes of the energy estimated.

2.2.3 Functional Aim of Energy Estimation Framework

As discussed above, the specific structural aims make the computation of energy possible, and then the target should be functionality of the framework at different phases.

Raw materials are first extracted from the earth in a form that can be easily stored and transformed. The raw material extraction phase can be an input to a new product life cycle. Therefore, it should be a starting point to estimate energy consumption during a product life cycle.

Design and manufacturing phase is the main concentration in this research. This phase determines the materials, shapes, and processing methods of final products and has a great impact of energy consumption during the whole life cycle. The energy estimated in manufacturing phase is complementary for design phase. It will address more attention to real manufacturing processes. Case studies for parts, assemblies and some simple industrial products will be conducted.

Situations vary for different products in use phase. If a product is energy consumptive during its life cycle, then the total energy consumed has to be taken
into account. This research will take an example of a product that are energy consumptive and estimate its energy consumption during the use phase.

When products come to their end of life, they need to be disposed or recycled. The energy consumed during this phase should also be considered.

2.3 Design and Development

2.3.1 Framework Construction

As aimed in the above sections, the framework should be expandable for developers and knowledge engineers and customizable for users. It should also be domain independent. In order to meet these requirements, the proposed framework for energy estimation should contain a knowledge data module, energy estimation modules and two interfaces, one each for knowledge engineer and user. A developer is also needed to implement various energy computational tools in different manufacturing domains and attach these tools to the energy estimation framework.

(a) Knowledge Data Module

In order to perform energy estimation, diverse knowledge data need to be acquired. These are material data, computational tool location data, manufacturing resource data, and manufacturing process data. All these data are stored in XML files that can be used to store and transfer knowledge. Accordingly, modifying data files will become convenient and the framework will be flexible for different manufacturing domains. (1) Material Data. The properties of materials are stored in material data. Different materials have different properties and they will have a great
impact on manufacturing processes. (2) Computational Tool Location Data. The data store locations of computational tools that are attached to the framework. These data will facilitate the framework to parse and invoke corresponding computational tools via user's selection. (3) Manufacturing Resource Data. Manufacturing resource data provide information of available manufacturing equipments for processing materials. Specifications of machines and cutting tools are included. (4) Manufacturing Process Data. The data store information for specific manufacturing processes such as milling and turning. It will help designers to choose feasible manufacturing processes during energy estimation process.

(b) Energy Estimation Modules

The module should perform two main functions: reading and parsing inputs for energy calculations and computing energy consumptions. In the inputs parsing module, all the XML files are parsed and geometric model of a product is obtained. Energy computation module will involve all kinds of detailed information including manufacturing methods and energy estimation formulas. Data related to geometric information, material properties, and manufacturing parameters will also be acquired to estimate energy consumptions.

(c) Interfaces

The framework consists of two interfaces. One is knowledge engineer interface which is responsible to provide interactions between knowledge
engineers and the framework. The other is user interface. The user interacts with different modules such as manufacturing process and energy computation through this interface.

The details of the proposed energy estimation framework are shown in Figure 4.

2.3.2 Characterization of Energy Estimation in Four Phases

Estimation of energy consumptions should start from the raw material extraction phase since it is an input to a new product life-cycle. The factors that are needed at this phase are listed in Table 2. Material type and extraction locations should be considered because the specific energy of the extracted materials varies for different materials and extracting locations. This research is
not aiming to provide complete data for all types of materials and locations all over the world. Sample calculations in U.S. for some typical materials will be conducted.

Table 2: Factors at Raw Material Extraction Phase

<table>
<thead>
<tr>
<th>Raw Material Extraction Phase</th>
<th>• Type of material</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Volume/weight of raw materials</td>
</tr>
<tr>
<td></td>
<td>• Specific energy depending on material type, extraction location, and type of processes used for extraction</td>
</tr>
</tbody>
</table>

Design and manufacturing phase determines the materials, shapes, and processing methods of final products and has a great impact of energy consumptions during the whole product life cycle. Thus, the main concentration lies in this phase. Please note that data for different manufacturing processes are far from complete. While based on the domain-independent feature, knowledge engineers from all over the world can add their knowledge to improve the framework. Completing designs for all manufacturing domains requires tons of efforts, and in this research, the manufacturing domain is limited as in machining and casting, and examples of turning, milling and casting will be taken.

The factors at design and manufacturing phase are shown in Table 3. In the design phase, products are analyzed without considering their manufacturing specifications. Such estimation will assist designers to improve their products regarding energy efficiency and help them in redesign process. Geometric information such as axis information, curvatures and face information will be abstracted using geometric kernel ACIS [54]. Such information is used to analyze products and automatically compute energy consumption. Mass properties such
as weight, surface areas, and volumes will also be obtained. These properties help to calculate removal volume of products. After collecting specific energy [55] for different manufacturing processes, by doing algebra of the specific energy and material removal volume, energy consumptions can be computed.

Table 3: Factors at Design and Manufacturing Phase

<table>
<thead>
<tr>
<th>Design Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric information of products</td>
</tr>
<tr>
<td>Axis info; external face info (determines the face type), curvature info (determines the direction and absolute value)</td>
</tr>
<tr>
<td>Mass property</td>
</tr>
<tr>
<td>Surface area</td>
</tr>
<tr>
<td>Weight</td>
</tr>
<tr>
<td>Specific Energy</td>
</tr>
<tr>
<td>Expandable panel: factors and different processes which can be expanded for future use</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manufacturing Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material: class, name, weight</td>
</tr>
<tr>
<td>Manufacturing processes (currently turning and milling):</td>
</tr>
<tr>
<td>Cutting speed, spindle speed, feed rate, depth of cut, width of cut</td>
</tr>
<tr>
<td>Tool/Machine parameter:</td>
</tr>
<tr>
<td>Type of machine</td>
</tr>
<tr>
<td>Tool diameter, length, angle</td>
</tr>
<tr>
<td>Time consumed: load, machined, idle</td>
</tr>
<tr>
<td>Material removal rate</td>
</tr>
<tr>
<td>Machined energy consumed</td>
</tr>
<tr>
<td>Total energy consumed</td>
</tr>
</tbody>
</table>

In the manufacturing phase, manufacturing specifications will be taken into account. Materials and details of manufacturing processes such as cutting speed, feed rate, and depth of cut, etc., will be considered. Parameters of tools and processing machines should also be utilized to perform the calculation. The energy consumed during assembling, machine loading, tool replacing and idle time should be reflected on the energy estimation process. Again, it is impossible
to collect all the data for different tools and machines, and the energy consumed during assembling, machine loading, tool replacing and idle time could be dramatically different due to the variety of manufacturers and workers, thus only some typical parameters will be taken to estimate energy consumptions.

Some products need power supply during their use phase. This phase can last for a certain period of time. Therefore, it is also one of the important phases in a product life-cycle where energy consumption should be considered. Since different products have their own specifications and life-spans, the energy estimation is considerably dependent on specific products. This research will take an example of a product that are energy consumptive and estimate its energy consumption during use phase. The factors include power and total time of using. Energy consumed in product’s supply chain such as transportation will not be considered in this research.

Table 4: Factors at Use Phase

| Use Phase | • Power needed to supply the products to function for their consumer  
| | • Time of using  
| | • Supply chain (not to be considered in this framework) |

When products come to their end of life, various operations might be taken such as reuse of certain parts, remanufacturing, recycling, and disposal. Due to the variety and complexity of reuse, remanufacturing and recycling, they are not considered in this research. This research will focus on disposal phase only during products’ end of life. Usually, energy consumption in disposal phase depends on the type of materials, specific energy during disposal.
Table 5: Factors at Disposal Phase

| Disposal Phase | • Type of materials  
|                | • Volume/weight of materials to be disposed  
|                | • Specific energy depending on material type, and disposal location |

In order to demonstrate the framework, a certain case studies will be conducted for energy estimations. Parts and assemblies will be analyzed through each phase of a product life-cycle. Estimation for simple industrial products will be made to show the validity of this framework. Energy consumed at each phase and for the whole life cycle will be obtained.

2.3.3 Use Cases

Figure 5 presents use cases for the proposed framework. The use cases illustrate the relationship between the framework and knowledge engineers and the relationship between the framework and energy estimate users.

Figure 5: Use Case Scenarios of Energy Estimation Framework
2.4 Integration and Test

Since the framework needs to interact with estimate users, knowledge engineers, XML files, and various energy estimation tools, integration is vital to make sure that the energy estimation framework works correctly. Figure 6 presents the integration among the framework, computational tools, knowledge engineers, and energy estimate users.

Figure 6: Integration among Framework, Computational Tools, Knowledge Engineer, and Estimate User

The framework itself can access XML files, so that it can read, parse, and write knowledge files for energy computation. Framework provides two interfaces which allow estimate users and knowledge engineers to interact with components of the framework. The framework is also capable to call specific
computational tools and receive the computational results as feedbacks. Knowledge engineers can load and edit XML to prepare energy estimation, and estimate users can use the tools provided by the framework to calculate energy estimation.

In SDLC, testing is important to confirm a robust and correct capability of software. At this point, all test cases are run to verify the correctness and completeness of the software. The testing plans for this energy estimation framework can be described as follows:

(a) Syntax checking

The software is checked to see if there are major functions and errors that will occur.

(b) Walkthrough

This step is to make sure that the basic structure and algorithm of the system is working correctly.

(c) Code inspection

In this part, the codes are reviewed for possible bugs or errors in coding technique.

(d) Module testing

Each module of the framework will be tested for possible errors, if errors occur in specific modules, code will need to be reviewed in terms of modules.
(e) Integration testing

The whole program will be tested. This step ensures the correctness of integration of each tool and modules of the framework.

(f) Test cases

Sample products and industrial cases will be taken as inputs to the framework to validate the framework.
CHAPTER THREE: ENERGY ESTIMATION FRAMEWORK

Based on the discussions in previous two chapters, energy estimation framework can be developed following SDLC steps. In this chapter, the overall design of the framework with its class diagram will be introduced first. Since knowledge management plays an important role in the framework, its mechanism will be illustrated afterwards. Then this chapter will present the overview of the energy estimation framework and elaborately illustrate its user interfaces and two modules: knowledge management module and energy estimation module.

3.1 Overall Design of Energy Estimation Framework

This research adopts Microsoft Visual C++ 2005, geometric kernel ACIS [54], Extensible Markup Language (XML), and the Unified Modeling Language (UML) to develop such energy estimation framework. All the codes are generated using VC++2005, and Microsoft Foundation Classes (MFC) is utilized to establish the software interface. XML is used to store and transfer knowledge, and the details of XML will be illustrated in the next section: Knowledge Management through XML. Geometric kernel ACIS is used for visualization and analysis of digital product’s data (CAD model). Object oriented approach is also adopted in developing the framework because this approach offers advantages of increased flexibility and reusability. In object oriented approach, class plays a critical role in the overall design. Class is an abstract data type, which groups a set of properties and operations together. A set of properties can be described as the state of an object or the instance variables of the object; while the behavior of an object is a set of operations. UML is a modeling tool for object oriented analysis.
and software design. It can be used for visualizing, constructing, and documenting the structure of a software system. In this research, UML is utilized to generate class diagram of the framework, which can explicitly represent the relationships between classes. The following UML class diagram describes the overall relationship between the classes that are implemented in this research.

![Class Diagram of Energy Estimation Framework](image)

Figure 7: Class Diagram of Energy Estimation Framework

The Doc-View structure is derived from MFC. When an application of the energy estimation framework is invoked, it creates an “EgyEstFwDoc” object which handles the opening and saving of documents. The Doc Object then creates the frame and invokes the view class of the energy estimation framework. In “EgyEstFrm” class, methods for menu, toolbar, and the active view have been
rewritten to decide the visibility and availability for relevant energy estimation components. “EgyEstView” class is responsible for the viewing of the framework, and it is also a key component where most of the energy calculations happen.

“EgyEstView” class manages energy estimation and knowledge communications. There are four classes that are aggregated to it: “CDialogClientType”, “CDialogUserInput”, “CDialogTool”, and “CDialogComptRslts”. Since there are two types of clients of the energy estimation framework: energy estimation users and knowledge engineers, and each of the clients are performing completely different tasks while using the framework. To avoid confusion, it is beneficial to setup different user interfaces and to allow only one interface functioning while using the framework. “CDialogClientType” class is responsible for dealing the client type control. Details of controlling client type can be found at §3.3: User Interfaces of Energy Estimation Framework. “CDialogUserInput” class parses and records user input data to provide parameters that might be required by specific energy computational tools. “CDialogTool” class parses locations of all computational tools that are attached to the framework, and execute the corresponding tool based on user’s request. “CDialogComptRslts” class parses and displays energy computational results back to energy estimation users. Implementation of these classes can be found at §3.5: Energy Estimation Module of Energy Estimation Framework.

Classes of “EgyEstToolExtr”, “EgyEstToolDM”, “EgyEstToolUse”, “EgyEstToolDsp”, “EgyEstToolPLC” are energy estimation tool classes that are
associated with the energy estimation framework class “EgyEstFwView”. They represent the computational tools that are attached to the framework. Each of the tools is responsible for calculating energy in extraction phase, design and manufacturing phase, use phase, disposal phase, and whole product life cycle respectively. After estimating energy, each of the tools is able to save the computational results in relative XML files. “CDialoMfg” and “GeomInfo” are two classes that are aggregated to “EgyEstToolDM” for assisting the estimation of energy in design and manufacturing phase. Details of implementing these classes can be found at both Chapter 3 and Chapter 4.

“CXML” is a class that can validate, parse, search, write and print XML file. This class is widely used in almost every component of the energy estimation framework and energy estimation tools. It plays a significant role in knowledge management. Details of this class will be illustrated in the following section: Knowledge Management through XML.

3.2 Knowledge Management through XML

3.2.1 XML and XML Schema (XSD)

An XML schema is a description of a type of XML document, typically described in terms of constraints on the structure and content of documents [56]. These constraints are generally expressed using some combination of grammatical rules governing the order of elements [56]. For example, Boolean type predicates that the content must satisfy to be a boolean variable. Data types govern the content of elements and attributes.
Since XML allows user-defined tags, it can be utilized to store knowledge. XML schema (XSD) provides a powerful method to access the structure and grammar of XML file. The data types of each elements have to be strictly defined in the XSD file.
Figure 8 shows the details of an XML file storing specific energy of different materials and its schema. Specific energy is considered to be an important property for energy calculations during implementation of the energy estimation software. Note that in the XSD file, the element type of “Name”, and “SpecE” have been set to “String” and “Float”. Such constraints provide possibility that the framework can be aware of which data type it is accessing.

3.2.2 Knowledge Management Class: CXML

In order to manage knowledge through XML, a class needs to be constructed which allows the framework and computational tools to utilize the knowledge stored inside XML files. The class diagram of the class is shown as Figure 9 below. “XML error handler” and “XML parser” are two private members inside the CXML class which can only be accessed by its member functions. These two members are responsible for XML parsing and error handling. “List of Elements”, “List of Text”, and “List of Attributes” variables can store the parsed information from an XML file. “XML Data Segment” is used to store any user input data. The class provides six operations to validate, parse, print, search, write, and obtain the information from an XML file.
3.2.3 Validation of XML

It is important to validate that the XML file matches its schema before parsing. The validation process utilizes the library provided by Xerces [57] with either the SAX (Simple API for XML) or the DOM (Document Object Model) parser. The difference between SAX and DOM parser is that SAX processes XML file in a sequential way spending only small chunk of memory, while DOM parses the XML file as a tree, which allows random access, but requires extra memory. For this research, it is preferred that the XML file can be parsed as a tree which allows random access. Therefore, DOM parser is adopted. To validate an XML file against an external schema, the following steps are implemented:

(a) Initialize and create an XML parser,
(b) Enable validation of the parser and register an error handler to receive notifications.

(c) Parse and validate XML file. Parsing of XML will be discussed below.

### 3.2.4 Parsing of XML

XML files store knowledge which can be utilized for energy estimation. In order to make the framework recognize knowledge, parsing of XML need to be conducted.

The algorithm of parsing an XML file can be described as follows:

(a) First, obtain the DOM tree and the root node of an XML file.

(b) Second, judge whether the tree is empty; if it’s not empty, iterate through all nodes, and then decide the node type, push the elements of the tree to different lists.

(c) At last, all the nodes have been stored to different lists according to the node types.

(d) Now XML parser can access any information from the XML file using “GetXML” method or print it out using “PrintXML” method.

Figure 10 below illustrates the algorithm of XML parsing in details.
3.3 User Interfaces of Energy Estimation Framework

As discussed in chapter 2, the framework is designed for both knowledge engineers and energy estimate users. Therefore, there are two separate interfaces available, one for knowledge engineers, and the other for energy estimate users. These interfaces are working independently, i.e. the clients have to identify which type they belong to before using the framework. Figure 11 illustrates the functionalities for each client types. The interface for knowledge engineers provides capability to edit XML files of different phases during a product life cycle, and to load and validate these XML files. The interface for energy estimation users provides methods to compute energy consumption during a product life cycle by invoking corresponding computational tools and
collecting user specified data. The interface also allows users to retrieve their computational results back.

![Diagram of interfaces for knowledge engineer and energy estimate user]

Figure 11: Structure of Interfaces of the Energy Estimation Framework

The implementation of these interfaces is based on Microsoft Foundation Classes (MFC). The basic idea is to create a client type control; before clicking the control and selecting client type, the corresponding functions are unavailable. When clicking the client type control, the framework prompts a dialog indicating selection of either a knowledge engineer or an energy estimation user. After selection, the framework will decide which functions are to be available.
The following figures show the results of the implementation of energy estimation framework.

Scenario 1: Open energy estimation framework

All the buttons on the framework have been set to be grey initially.

![Figure 12: Scenario 1: Open Energy Estimation Framework](image)

Scenario 2: Click new or open a part file

The framework is created using an ACIS project. After clicking new or opening a part file, the user interface for the energy estimation framework will be displayed. The corresponding menu and toolbar for both energy estimate user and knowledge engineer have been set to be grey. The estimate client control is now available to decide the client type. Figure 13 shows the interface of the energy estimation framework. It includes a client control component, the interface for energy estimate users, and the interface for knowledge engineers.
Figure 13: Interface of the Energy Estimation Framework

Figure 14 shows the result of Scenario 2 when clicking new or open a part file.

Figure 14: Scenario 2: Click New or Open a Part File
Scenario 3: Select client type

Client type can be selected by clicking the client control button. A dialog is popping out to decide whether the client is an energy estimate user or a knowledge engineer. After selection, the corresponding menu and toolbar will be activated. It is important to make the interface for energy estimate user and knowledge engineer independent to avoid changing knowledge file while estimating energy consumption. These independent interfaces make the framework robust, easy to manage, and have a clearer structure and data flow. Figure 15 shows the result of Scenario 3 for client type selection.

Figure 15: Scenario 3: Client Type Selection
Figure 16 and Figure 17 show scenarios for selecting “For Energy Estimation User” and “For Knowledge Engineer” respectively.

![Energy Estimation Framework](image)

**Figure 16: Scenario 3 (a): Client Type Selected as Energy Estimation User**

By selecting “For Energy Estimation User”, the energy estimate user interface is activated. The detailed functions for energy estimation will be illustrated in section 3.4: Energy Estimate Module of Energy Estimation Framework.
By selecting “For Energy Estimation User”, the knowledge engineer interface is activated. The detailed functions for knowledge management will be illustrated in section 3.3: Knowledge Management Module of Energy Estimation Framework.

3.4 Knowledge Management Module of Energy Estimation Framework

The core function of this module is to manage knowledge. In this research, it is decided that knowledge is stored and transformed using XML. The mechanism for knowledge management has already been introduced in section 3.1. In knowledge management module, the software provides entries for selecting knowledge file that will be used in this framework.
Figure 18: Menu and Toolbar of Knowledge Management Module

Figure 19: Functions in Knowledge Management Module
As is shown in Figure 18 and Figure 19, the framework provides XML file entries for materials extraction phase, design and manufacturing phase, and disposal phase. By clicking on each entry, knowledge engineer will be able to open, edit, and save XML files. Please note that in use phase, there is no XML file necessary, because the use phase energy consumption is determined by product’s power and life span, these parameters should be specified by users. “Load XML” is one of the most important features of the framework, by which the framework itself can load, read, validate, and parse XML file. The framework can thus retrieve useful information for energy estimation and decide which computational tools to need to be invoked for the calculation.

The following figures show scenarios of using the knowledge management module. Figure 20 and Figure 21 show scenarios of configuring knowledge files (XML) before calculating energy consumptions.

Figure 20: Scenario of Configuring XML File in Knowledge Management Module: Open
In Figure 20, after clicking on each entry, a dialog will pop out asking knowledge engineer to select corresponding XML file. Figure 21 shows the scenario for editing and saving a sample XML file. In this scenario, if an XML file that will be used in the energy estimation process is selected, it will be displayed in a separate window. The knowledge engineer can now edit, expand, and save the XML file. The information in the XML file can then be used for energy computation.

Figure 21: Scenario of Configuring XML File in Knowledge Management Module: Edit and Save
Figure 22 presents another scenario of “Load XML” while using the knowledge management module. The dialog in the figure shows the parsing results of previous XML file shown in Figure 21. Note that this dialog also contains a “ValidateXML” button to validate the schema of XML file.

![Energy Estimation Framework - [BASESAT(openg)]](image)

**Figure 22: Scenario of Using Knowledge Management Module: Load XML**

### 3.5 Energy Estimation Module of Energy Estimation Framework

The core of this module is to estimate energy consumption of different phases during a product life cycle. Figure 23 shows the menu and toolbar for energy estimation module.
Energy estimation module mainly contains three components: user input, energy estimation tools, and computational result.

(a) User input option asks energy estimation users to provide information that has to be specified by themselves for energy calculation.

(b) Energy estimation tools compute energy consumption during a product life cycle. They are used to calculate energy in materials extraction phase, design and manufacturing phase, use phase, disposal phase, and whole product life cycle.

(c) Computational results display the outcomes for energy estimation.
Figure 24 shows the components in energy estimation module. As is part of the framework, scenarios for user input option and computational results will be provided here, while details of each energy estimation tools will be illustrated in Chapter Four: Computational Tools of Energy Estimation Framework.

User input option is used for collecting user specified information, for example, in the scenario below, user can specify materials as their input from the list. The list is created dynamically based on the elements in the XML file. The XML file used in this example can be found in Figure 21. This user input information will also be stored as a separate XML file which will be read by relevant computational tools. The details of communication between XML, framework, and computational tools will be discussed in chapter five.
Energy estimation tools for different phases of a product life cycle are integrated in the module. Clicking relevant button will invoke corresponding energy computational tool. These tools will perform energy estimations and output results into separate XML files. The details of computational tools will be discussed in the next chapter.

After calculating energy consumptions using energy estimation tools, users are able to retrieve information from the computational result component. This component is responsible for displaying previous computational result. Figure 26 shows a scenario of retrieving computational results after calling energy estimation tools. The details of interaction between tools, XML, and framework will be illustrated in further chapters.
Figure 26: Scenario of Using Energy Estimation Module: Computational Result
CHAPTER FOUR: COMPUTATIONAL TOOLS OF ENERGY ESTIMATION

FRAMEWORK

The computational tools attached to the framework can be used to calculate energy consumptions during material extraction phase, design and manufacturing phase, use phase, disposal phase, and whole product life cycle.

![Diagram of Energy Estimation Framework]

Figure 27: List of Tools that are attached to Energy Estimation Framework

These energy computational tools can be attached to the framework and expanded. According to the design analysis in chapter 2, computational tools in material extraction, use, disposal, and PLC do not require domain knowledge to develop, i.e., they are almost uniform for all manufacturing processes. Therefore, in order to make the framework be an integrated piece of all phases during a product life cycle, tools for extraction, use, disposal, and PLC are designed to be built-in in the framework, yet they can also be developed separately and attached to the framework if needed. Tools for design and manufacturing phase require
domain knowledge to develop, i.e., energy in different manufacturing processes is calculated in a completely different manner. Thus, design and manufacturing energy estimation tools have to be separate from the framework, and attached to the framework through an executable file. §4.2 will introduce an XML file for storing all the computational tools’ locations and illustrate how to implement the separate tool and attach the tool to the framework. The way to implement a tool separately and attach it to the framework can be recognized as a generic method to develop tools in specific domains and integrate with the framework. Such implementation ensures the framework to satisfy the flexible and expandable requirements. This chapter will introduce the computational tools listed in Figure 27 in details.

4.1 Materials Extraction Phase Energy Estimation Tool

According to the discussion in Chapter 2, calculating energy in materials extraction phase requires parameters of material type, specific energy during extraction, and the amount to be extracted. Parameters of material type and the amount to be extracted are prompted for user input. Some of the specific energy during extraction phase is provided in [58]. Figure 28 shows the tool to calculate energy consumption during materials extraction phase.
The following equation shows the method for calculating energy during extraction phase. It can be simply expressed as:

\[ E_{ext} = SpecE_{ext} \times M_{ext} \quad (1) \]

Where \( E_{ext} \) stands for energy during extraction, \( SpecE_{ext} \) is specific energy during extraction, and \( M_{ext} \) is the amount to be extracted.

In Figure 29, the list of materials is parsed based on the XML file. The framework can parse the corresponding XML file input or edited by knowledge engineer, and display the list of materials dynamically. This dynamic behavior ensures the framework to meet the customizable and expandable requirements.
After calculating the extraction energy, the framework will pop out a message box to provide the computational results to the user and record the result into a separate XML file for later use. The use of the separate XML file will be illustrated in Chapter 5.

4.2 Design Phase Energy Estimation Tools

4.2.1 Overview of Energy Estimation in Design and Manufacturing Phase

In order to compute manufacturing energy of a product, specific domain knowledge related to the manufacturing process is required. For example, machining and casting are two different manufacturing methods; energy consumed in manufacturing a part through machining is computed in a completely different manner than through casting. Therefore, in order for the framework to be domain independent and expandable, the energy estimation
tools for design and manufacturing phase have to be separate from the framework. Different tools can be developed as separate executable files by developers who have relevant domain knowledge, and then these tools can be attached to the framework. The locations of the computational tools are stored in an XML file and the content of this XML file can be added, modified, and expanded following the schema files predefined by this research. Locations of computational tools indicate where the executable files of computational tools lie. The framework is able to parse this XML file, retrieve tool paths, and call relevant tools based on the parsed tool paths. As stated in the system boundaries in Chapter 2, this research will only demonstrate a development of design and manufacturing tools for estimating energy during machining and casting operation.

Figure 30: Computational Tool Selection
In this research, the tools developed to estimate energy in design and manufacturing phase are named as “DMCasting” and “DMMachining” as shown in Figure 30. The framework will ask user to select a computational tool, and then an executable file for selected tool will be invoked. Figure 31 shows the tool to calculate machining energy consumption during design and manufacturing phase. In design phase, the tool gives user options on whether to automatically compute the stock volume. If there are existing stocks to manufacture a product, user can input the stock information rather than using the tool itself to calculate the stock.

![Figure 31: Energy Estimation Tool: Design and Manufacturing Phase](image)

In design phase, the estimation will mainly concentrate on the analysis of a product’s geometric information. While in manufacturing phase, it needs to take into account many details of manufacturing processes and plans. At design stage, there will be a range for the estimated manufacturing energy consumption. An
accurate number for energy with very little uncertainty can only be obtained when
details of the manufacturing process plans are available. With lack of process
plans at initial stages of design, only a range of energy estimates can be
computed. The large range is attributed to the manufacturing variations
associated with different tools, fixtures, lubricants, machines, operators, etc. The
relevant information regarding these manufacturing specifics is not available at
early design stages and will be accounted in manufacturing stages. Correspondingly the range of energy will be refined as the design progresses
through manufacturing phase. Although, these initial estimates have large ranges,
the information is relevant in selecting an appropriate energy efficient design
from two or more designs that have disparate energy ranges. The range of
estimated energy in design and manufacturing phase is shown in Figure 32.
4.2.2 Design Energy Estimation for Milling Operation

The following two sections will discuss design energy estimation for milling and turning operation respectively. In design phase, the geometry of individual parts and assembly are used to compute the manufacturing energy, subsequently useful for decision making at the preliminary/embodiment and detail design stages.

4.2.2.1 Steps to Compute Energy Estimates for Milling Operation

Figure 33 provides the details for computing such energy estimates for parts and assemblies. Assembly or part level geometric data with assigned material properties forms the input of estimation. The next step is the computation of the minimum oriented bounding box for each part. Here it is assumed that the minimum oriented bounding box would serve as a stock for machining the final part. A minimum oriented bounding box is necessarily identified for the following two reasons:

a) The orientation of a part in an assembly may not be aligned with the coordinate planes

b) It is aimed to automate the computation of stock in design phase.

The next step is to identify the removal volume. Removal volume can be identified by performing Boolean subtraction between each part and respective stock (minimum oriented bounding box). The volume of the resultant geometry from Boolean subtraction will be the removal volume for manufacturing (milling in our case) the part from the computed stock. We then compute the machining energy for the part based on generalized data available in [55]. The machining
energy for all the parts is later aggregated to arrive at the overall energy for machining the whole assembly. In this step, information regarding the energy intensive parts is also provided in order to aid energy efficient redesign.

Figure 33: Overall Algorithm for Computing Energy and CW for Part and Assembly in Design Phase for Milling Operation

4.2.2.2 Algorithm for Calculating the Minimum Oriented Bounding Box

Various researchers have presented algorithms for computing minimum oriented bounding box [59-62]. A survey and classification of different kinds of box fitting algorithms is provided by [63-65]. The algorithm presented here takes a novel approach as it utilizes the minimum zone algorithm used in verifying flatness tolerances during inspection. It must be noted that we are still to test the
comparative efficiency of the proposed algorithm. The related discussion is not the focus of current research.

Figure 34 presents the algorithm in detail. The first step is to create points on the part geometry. Using these points, a convex hull for the part is computed using qhull [66]. Utilizing the facets and vertices of the convex hull, a minimum zone is computed using the min-max algorithm. This algorithm generates two parallel planes that are at a minimum distance apart and all the points of the convex hull lie between them. After the minimum zone is obtained, all the points of the convex hull are projected on to one of the planes of the minimum zone. Then the plane is transformed to the x-y plane. The two dimensional convex hull is then computed, followed by two minimum zones where the product of the distances is minimized. The two dimensional minimum zones are then transformed back to their original location, using inverse transform. Finally, the resultant minimum oriented bounding box is obtained from these three minimum zones.
Figure 34: Algorithm to Compute the Minimum Oriented Bounding Box

To test the above algorithm for minimum oriented bounding box, a revolved part (as it does not have planar faces) is used as shown in Figure 35(a). The input to the convex hull consists of 2566 vertices on the geometry while the final convex hull consists of 586 vertices and 1139 facets. The first minimum zone is found for planes with its normal \((-0.508751, 0.43743, -0.741503)\) and distance between them as 82.3878.
Figure 35 (a) Initial Revolved Part as an Input to the Minimum Oriented Bounding Box Algorithm, (b) XZ Cut View of the Overlapping Minimum Oriented Bounding Box and the Revolved Part.

All the vertices of the 3D convex hull are projected onto this plane. The 2D convex hull of these projected vertices consists of 44 vertices and 44 facets (edges) only. Using the 2D convex hull, two perpendicular minimum zones are computed where the product of the distances is minimized. Utilizing the three minimum zones, the minimum oriented bounding box is computed for the given part. Figure 35 (b) shows a cut view of the fitting of the revolved part (Figure 35 (a)) and the computed minimum oriented bounding box.

After the vertices are projected onto the 2-D, the minimum zone algorithm alone cannot be used. The minimum zone algorithm searches for the smallest distance between two parallel planes within which all the points should lie.

Figure 36: (a) Initial Part as an Input to the Minimum Oriented Bounding Box Algorithm, (b) The Convex Hull of the Part Computed Using Qhull
Figure 37: (a) YZ Cut View of Bounding Box Using a Minimum Zone that Has the Minimum Distance for the 2D Projected Vertices (b) YZ Cut View of the Bounding Box Using the Minimum Zone Obtained by Minimizing the Product of the Distances of All Possible Minimum Zones for the 2D Projected Vertices.

The current algorithm searches two minimum zones with minimized product of the minimum distances. The example part shown in Figure 36(a) will be used to demonstrate the difference in two algorithms. Figure 36(b) shows the convex hull generated using qhull. Figure 37(a) and (b) shows the overlap of the resulting bounding box and part from Figure 36(a). The bounding box is computed with the minimum zone algorithm. The minimum zone algorithm found $x_1$ to the minimum distance between two planes within which all the 2D projected points of the part lies. But the consequence of selecting $x_1$, is the distance $y_1$ to form the box around the part. The bounding box in Figure 37(b) is computed using the minimized product of two minimum distances. In other words the algorithm searches for a minimum value for $x*y$. This leads to the rectangle with lengths $x_2$ and $y_2$. It is intuitively clear that the minimum oriented bounding box is the one in Figure 37(b).
4.2.2.3 Case Studies for Energy Estimation in Design Phase for Milling Operation

This section will present case studies for energy estimation in design phase for milling operation from two view points, single part and an assembly of parts. Energy and CW estimation for each case will be presented for the preliminary design and detail design. CAD file handling and visualization is created using ACIS kernel.

Case 1: Single Part: Cover Plate

The sample part is a cover plate shown in Figure 38. Shaded view of the cover plate in the preliminary and detail design stages are shown in Figure 39. In this case study, the only difference between the preliminary and detail design are the fillets at sharp corners in the cover plate. The material of the cover plate is cast iron. Convex hull and the computed minimum oriented bounding box fitting the cover plate are shown in Figure 38.

![Figure 38: (a) Convex Hull for the Cover Plate (b) Fit of the Minimum Oriented Bounding Box and the Cover Plate.](image)

Utilizing the machining energy data from [55], energy estimates for milling the cover plate can be computed as shown in Table 6. The specific energy for cast iron is 1.6 to 5.5 W-s/mm³. Specific energy is given as large range, because
of the manufacturing variations associated with different tools, fixtures, lubricants, machines, operators, etc. The CW can be computed from Watt-hour units of energy consumption based on the data from [67]. Computations of CW in Table 6 are performed assuming Maryland\(^1\) (factor \(f = 0.62\) Metric Tons/MWh) as the state in which manufacturing is being performed.

Table 6: Comparison of the Calculations of Range of Energy and CW Estimates of the Part at the Preliminary and Detail Design Stages

<table>
<thead>
<tr>
<th>Preliminary design stage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Part volume</td>
<td>25243.14 mm(^3)</td>
</tr>
<tr>
<td>Stock volume</td>
<td>72115.42 mm(^3)</td>
</tr>
<tr>
<td>Removal volume</td>
<td>21331.85 mm(^3)</td>
</tr>
<tr>
<td>Milling energy estimation</td>
<td>9.48-32.59 (Watt-hour)</td>
</tr>
<tr>
<td>CW estimation</td>
<td>5.87-20.20 (10^{-6}) metric Tons</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detail design stage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Part volume</td>
<td>25540.42 mm(^3)</td>
</tr>
<tr>
<td>Stock volume</td>
<td>72115.42 mm(^3)</td>
</tr>
<tr>
<td>Removal volume</td>
<td>21036.04 mm(^3)</td>
</tr>
<tr>
<td>Milling energy estimation</td>
<td>9.34-32.13 (Watt-hour)</td>
</tr>
<tr>
<td>CW estimation</td>
<td>5.79-19.92 (10^{-6}) metric Tons</td>
</tr>
</tbody>
</table>

From the results of the computation, it is evident that the addition of fillets on the cover plate has decreased the removal volume for manufacturing the cover plate. Fillets at certain locations increase the removal volume while at other locations, they decrease the removal volume. For reducing the energy and carbon weight, the radius of fillets at convex corners/edges should be kept as small as possible, while the radius of the fillets at the concave corners/edges should be increased. Since in this simplified case, we only considered fillets as the change between preliminary and detail design and hence suggestions for

---

\(^1\) Different states have different electricity to emissions ratio based on the fact that electricity is supplied in different states from different sources, such as hydro, nuclear, coal, etc. Some of these have CO\(_2\) emissions associated with them.
reducing energy or carbon weight can only be made based on the geometry of fillets.

Figure 39: Effect of Fillet in Removal Volume for Convex and Concave Edges.

**Case 2: Assembly: Slider Crank Mechanism**

A simplified slider-crank assembly case is shown in Figure 40. The difference between preliminary and detail design stages are again at the level of fillets at the sharp corners. The material for the parts is assumed to be cast iron.

Figure 40: Assembly Case: Slider-crank Mechanism

Computing the removal volume for individual parts is handled as described in previous section. After the range of machining energy is estimated
for each part, worst-case and statistical-case energy ranges are computed for the assembly.

For computing the worst-case and the statistical-case energy ranges, principles from tolerance analysis are borrowed. For further details regarding the tolerance principles for worst-case and statistical-case analysis please refer to [68, 69]. The manufacturing energy of the assembly can be computed as the sum of manufacturing energies of individual parts and the energy for assembling all the parts together, as shown in equation (2).

\[
E_{assembly} = \sum_{i=0}^{n} E_{part-i} + \sum_{i,j,k,...,m=0}^{n} E_{assem-ijk...m} \\
= \sum_{i=0}^{n} (RV_{part-i} \times SE_{part-i})
\]  

where \( E_{assembly} \) represents total manufacturing energy of the assembly, \( E_{part-i} \) represents manufacturing energy of the part \( i \) and \( E_{assem-ijk...m} \) represents energy for assembling all the parts, \( RV_{part-i} \) represents removal volume for part \( i \), and \( SE_{part-i} \) represents specific energy for part \( i \). For the purposes of simplifying the calculations, in this paper we have assumed that the assembling energy is zero. Therefore the maximum and minimum manufacturing energy for the assembly can be computed as shown in equations (3) and (4) respectively.

\[
E_{assembly-max} = \sum_{i=0}^{n} E_{part-max-i} \\
= \sum_{i=0}^{n} (RV_{part-i} \times SE_{part-max-i})
\]  

72
\[ E_{\text{assembly-min}} = \sum_{i=0}^{n} E_{\text{part-min-i}} = \sum_{i=0}^{n} (RV_{\text{part-i}} \times SE_{\text{part-min-i}}) \] (4)

where \( E_{\text{assembly-max}} \), \( E_{\text{assembly-min}} \), \( E_{\text{part-max-i}} \), \( E_{\text{part-min-i}} \), etc., represent manufacturing energy maximum and minimum for the assembly and \( \text{part-i} \), respectively. Now, the manufacturing energy variations (\( \Delta E_{\text{assembly}} \)) of the assembly can be computed as shown in equation (5).

\[ \Delta E_{\text{assembly}} = E_{\text{assembly-max}} - E_{\text{assembly-min}} = \sum_{i=0}^{n} \Delta E_{\text{part-i}} = \sum_{i=0}^{n} E_{\text{part-max-i}} - E_{\text{part-min-i}} = \sum_{i=0}^{n} (RV_{\text{part-i}} \times \Delta SE_{\text{part-i}}) \] (5)

Table 7 shows the calculation in detail. The second column lists the removal volume for each part. The third column provides the range estimate of machining energy for each part of the assembly shown in Figure 40. For machining energy estimates for the whole assembly, two methods are employed. In the worst-case analysis, minimum and maximum of the range of energy for each part are respectively added together.

For statistical analysis, one could assume that the energy estimates with respect to the practical machining energy would be accurate within \( 3\sigma \) to \( 6\sigma \) variations. The results for statistical case includes mean (\( \mu \)) and standard deviation (\( \sigma \)) of the energy estimates for the whole assembly. As before, the mean and standard deviation of the assembly manufacturing energy \( \mu_{E_{\text{assembly}}} \) and \( \sigma_{E_{\text{assembly}}} \) can be computed from the mean and standard deviation of the
individual part manufacturing energy $\mu_{E-part-i}$ and $\sigma_{E-part-i}$ as shown in equations (6) and (7), respectively.

$$\mu_{E-assembly} = \sum_{i=0}^{n} \mu_{E-part-i}$$  \hspace{1cm} (6)

$$\sigma_{E-assembly}^2 = \sum_{i=0}^{n} \sigma_{E-part-i}^2$$  \hspace{1cm} (7)

where, $\mu_{E-assembly}$, $\mu_{E-part-i}$, $\sigma_{E-assembly}$ and $\sigma_{E-part-i}$, represents mean and standard deviations of the manufacturing energy of the assembly and the part, respectively.

$$6\sigma_{E-part-i} = \Delta E_{part-i}$$  \hspace{1cm} (8)

The results of these statistical CW computations are also shown in Table 7.

Table 7: Comparison of the Calculations of Range of Energy and CW Estimates of the Part at the Preliminary and Detail Design Stages

<table>
<thead>
<tr>
<th>Preliminary design stage</th>
<th>Removal volume (mm³)</th>
<th>Energy estimates (Wh)</th>
<th>Carbon weight (10⁻⁶ metric Tons)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crank</td>
<td>34532.47</td>
<td>15.34-84.41</td>
<td>9.51-52.33</td>
</tr>
<tr>
<td>Rod</td>
<td>109631.00</td>
<td>48.72-267.98</td>
<td>30.20-166.15</td>
</tr>
<tr>
<td>Slider</td>
<td>1608.73</td>
<td>0.71-3.93</td>
<td>0.44-2.43</td>
</tr>
<tr>
<td>Base</td>
<td>178648.93</td>
<td>79.39-436.69</td>
<td>49.22-270.07</td>
</tr>
<tr>
<td>Assembly Worst Case</td>
<td></td>
<td>144.18-793.02</td>
<td>89.39-491.67</td>
</tr>
<tr>
<td>Statistical Case ($\mu, \sigma$)</td>
<td></td>
<td>468.60; 70.81</td>
<td>290.53; 43.90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detail design stage</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Crank</td>
<td>34994.95</td>
<td>15.55-85.54</td>
<td>9.64-53.03</td>
</tr>
<tr>
<td>Rod</td>
<td>110039.99</td>
<td>48.90-268.98</td>
<td>30.32-16.67</td>
</tr>
<tr>
<td>Slider</td>
<td>2063.49</td>
<td>0.91-5.04</td>
<td>0.56-3.12</td>
</tr>
<tr>
<td>Base</td>
<td>180932.04</td>
<td>80.41-442.27</td>
<td>49.85-274.213</td>
</tr>
<tr>
<td>Assembly Worst Case</td>
<td></td>
<td>145.79-801.85</td>
<td>90.39-497.14</td>
</tr>
<tr>
<td>Statistical Case ($\mu, \sigma$)</td>
<td></td>
<td>473.82; 71.54</td>
<td>293.77; 44.36</td>
</tr>
</tbody>
</table>

From the worst-case analysis, the percentage contributions for Crank, Rod, Slider and Base are 10.63, 33.79, 0.49 and 55.06, respectively. The percent contribution suggests that any reduction in energy for machining the Base will
have more impact on the overall energy than the Slider. If 10% reduction is desired in the machining energy, then the Crank, Rod, Slider and Base need to be redesigned so that the machining energy reduces by 1.06, 3.37, 0.0 and 5.5 percent, respectively.

4.2.3 Design Energy Estimation for Turning Operation

4.2.3.1 Steps to Compute Energy Estimates for Turning Operation

Figure 41 provides the details for computing such energy estimates for parts and assemblies. Assembly or part level geometric data with assigned material properties forms the input for energy estimation. The next step is to identify turnable faces of parts and assemblies and compute the minimum turnable entities. A minimum turnable entity is necessarily identified for the following two reasons: a) the parts may not be pure turnable and it may need other operations such as milling. Therefore, only turnable removal volume is useful for computing turning energies. b) The energy estimation tool aims to automate the computation of bounding stock in design phase. It is assumed that the minimum bounding cylinder would serve as a stock for machining the final part. The next step is to compute the minimum bounding cylinder for each part and calculate the removal volume. Removal volume can be identified by performing Boolean subtraction between each turnable entity and respective stock (minimum bounding cylinder). The volume of the resultant geometry from Boolean subtraction will be the removal volume for manufacturing (turning in our case) the part from the computed stock. Then the machining energy for the part can be computed based on generalized data available in [55].
Figure 41: Overall Algorithm for Computing Energy and CW for Part and Assembly in Design Phase for Turning Operation

The machining energy for all the parts is later aggregated to arrive at the overall energy for machining the whole assembly. In this step, information regarding the energy hogging parts is also provided in order to aid energy efficient redesign.

4.2.3.2 Algorithm for Calculating the Removal Volume

Before calculating the removal volume, the turnable and unturnable faces have to be discovered. If it is a pure turnable part, the removal volume can be
directly obtained by subtracting the volume of the part from the minimum bound cylinder. However, in most cases, parts cannot be produced by only using turning operations, the algorithm presented in this paper takes a novel approach to automatically recognize turnable/unturnable faces and calculate a minimum turnable entity which allows turning machine to fabricate at the most extent. It must be noted that we are still to test the comparative efficiency of the proposed algorithm. The related discussion is beyond the scope of this research. Figure 42 presents the algorithm in detail.

The first step is to get part geometric data and acquire the rotation axis. The rotation axis is also the major turning axis. Note that in this research, it is assumed that parts have only one turning axis. After attaining turning axis, recognition of turnable/unturnable faces is necessary. Turning operation is widely used in machining process that is described as removal of material from the surface diameter of a rotating work piece with a single-point tool. Turning is a combination of two movements: rotation of the work piece and feed movement of the cutting tool [70]. The faces on a part machined by turning are represented as revolved faces in solid modeling.
Figure 42: Algorithm to Compute the Removal Volume

Figure 43 shows some examples of turnable surfaces. The turnable faces in Figure 43 (a), (b), and (c) could be produced by longitudinal turning, angle cutting, and facing respectively [70]. In this research, a turnable entity is defined as a continuous planar profile that consists of one or more bounded curves, an axis of revolution, and the material direction. A bounded curve is a curve with a
start and end parameters that specify its bounds [71]. A turnable entity must be composed of one or multiple revolved faces.

Figure 43: Example of Turnable Faces: (a) Cylindrical Face; (b) Conical Face; (c) Planar Perpendicular to Rotation Axis;

In this research ACIS geometric kernel is utilized to build algorithm and analyze parts and assemblies. Identification of certain simple geometrical features, such as cylindrical, conical and planar faces, can be performed.

From the previous step, whether the part is pure turnable or not has already been determined. If it does have unturnable features, the next step is circumferential projection. The concept of circumferential projection (CPJ) is proposed to implement this approach [72]. To obtain the CPJ of an entity, all points of the entity are rotated about an axis onto a single plane on which the axis lies, and the union of all these coplanar rotated points is the CPJ of the entity. Figure 44 illustrates the CPJ of several types of entities.
Figure 44: Circumferential Projections of Various Entities [72]

Figure 44 (a) shows the CPJ of a single point. Figure 44 (b) displays the CPJ of an edge. Figure 44 (c) indicates the CPJ of a revolved face $f$ whose axis is collinear with the global $Z$-axis. The CPJ of $f$ is an edge $e$, which can also be viewed as the CPJ of all the edges on the external loop of the face [72]. Obviously, revolving CPJ could return the part to the minimum turnable entity regardless of unturnable faces.

Figure 45 shows an example of a mill/turn part. Notice that for this part face 4 is unturnable, which means that the part needs to be manufactured by at least two methods, i.e. turning and milling. If one uses normal projection, there will be mistakes in calculating the radius of the bounding cylinder. Figure 45 (b) shows the vertical projection of the part. The actual revolved radius is 20 rather than 14.120 that is obtained by vertical projection. By using circumferential projection, the actual revolved radius could be found as shown in Figure 45 (c). After circumferential projection, the expected profile is obtained; revolving the profile along the rotation axis will build a minimum pure turnable entity. This pure turnable part will be utilized to calculate the minimum bounding box.
After identifying minimum turnable entity, the next step is the computation of minimum bounding cylinder. In previous section for milling energy estimation, the research presents an algorithm to calculate the minimum oriented bounding box using minimum zone and convex hull and transfer points from 3D-2D-3D. Using this method, minimum oriented bounding box could be obtained. Since the bounding box is calculated based on a revolved part (minimum turnable entity), the bounding cylinder can then be easily computed. Removal volume will be determined by using Boolean operations.
Figure 46: (a) Turnable Part; (b) Wireframe Showing the Overlap of the Revolved Part and its Minimum Oriented Bounding Box; (c) Wireframe Showing the Overlap of the Revolved Part and its Minimum Bounding Cylinder; (d) Cut View Showing the Overlap of the Revolved Part and its Minimum Bounding Cylinder

Figure 46 below shows the results of calculating the minimum bounding cylinder, Figure 46 (a) is the original turning part and (b) calculates a minimum bounding box outside, (c) and (d) show the wire frame and cut view of the overlapping of the turning part and its minimum bounding cylinder.

4.2.3.3 Case Studies for Energy Estimation in Design Phase for Turning Operation

This section will also present case studies for energy estimation in design phase for turning operation from two view points, single part and an assembly of parts. Energy and CW estimation for each case will be presented for the
preliminary design and detail design. Notice that in design phase, only a range of machining energy can be estimated to assist in choosing a low manufacturing energy design. The actual energy used to manufacture the parts, based on tool types, coolant, temperature, fixture, setups and idling energy, etc., will be discussed in next section: energy estimation in manufacturing phase. CAD files handling and visualization is created using ACIS kernel.

**Case 1: Energy Estimation for Single Part**

The sample is a mill/turn mixed part shown in Figure 47. It has four unturnable slots which need to be produced by milling operation. Both shaded and wireframe of the part in the preliminary and detail design stages are shown below.

![Figure 47: (a) Solid and (b) Wireframe View of the Part at the Preliminary Design Stage. (c) Solid and (d) Wireframe View of the Part at the Detail Design Stage](image)

Figure 47: (a) Solid and (b) Wireframe View of the Part at the Preliminary Design Stage. (c) Solid and (d) Wireframe View of the Part at the Detail Design Stage
In this case study, the material is cast iron. Figure 48 (a) and (b) show the computation results of the minimum turnable entity. Figure 48 (c) and (d) show the results of computing its minimum bounding cylinder.

Utilizing the machining energy data from [55], energy estimates for turning the part can be computed as shown in Table 8. The specific energy for cast iron is 1.6 to 5.5 W-s/mm³. The CW can be computed from Watt-hour units of energy consumption based on the data from [67]. Computations of CW in Table 8 are
performed assuming Maryland (factor $f = 0.62$ MetricTons/MWh) as the state in which manufacturing is being performed.

Table 8: Comparison of the range of energy and CW estimates of the part at the preliminary and detail design stages

<table>
<thead>
<tr>
<th>Preliminary design stage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Part volume</td>
<td>27740893.74 mm$^3$</td>
</tr>
<tr>
<td>Stock volume</td>
<td>49085905.25 mm$^3$</td>
</tr>
<tr>
<td>Turnable part volume</td>
<td>29827851.45 mm$^3$</td>
</tr>
<tr>
<td>Removal volume</td>
<td>19259501.53 mm$^3$</td>
</tr>
<tr>
<td>Turning energy estimation</td>
<td>8559.78-29424.24(Watt-hour)</td>
</tr>
<tr>
<td>CW estimation</td>
<td>5.17-18.93($10^{-3}$ metricTons)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detail design stage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Part volume</td>
<td>27603235.44 mm$^3$</td>
</tr>
<tr>
<td>Stock volume</td>
<td>49085905.25 mm$^3$</td>
</tr>
<tr>
<td>Turnable part volume</td>
<td>29677150.58 mm$^3$</td>
</tr>
<tr>
<td>Removal volume</td>
<td>19410311.83 mm$^3$</td>
</tr>
<tr>
<td>Turning energy estimation</td>
<td>8626.80-29654.64(Watt-hour)</td>
</tr>
<tr>
<td>CW estimation</td>
<td>5.22-19.54($10^{-3}$ metricTons)</td>
</tr>
</tbody>
</table>

From the results of the computation, estimation of energy and CW consumption has been made in design stages. Decisions regarding energy saving and pollution reduction could then be conducted. It is evident that the addition of fillets on the part has decreased the removal volume for manufacturing. Fillets at certain locations increase the removal volume while at
other locations, they decrease the removal volume. For reducing the energy and carbon weight, the radius of fillets at convex corners/edges should be kept as small as possible, while the radius of the fillets at the concave corners/edges should be increased. Since in this simplified case, we only considered fillets as the change between preliminary and detail design and hence suggestions for reducing energy or carbon weight can only be made based on the geometry of fillets.

**Case 2: Energy Estimation for Assemblies**

![Figure 49: (a) An Assembly of Parts (b) Assembled Model](image)

(c)Cut View Showing the Overlap of Assembled Parts and its Bounding Cylinder (d) Wireframe View Showing the Overlap of Assembled Parts and its Bounding Cylinder
A simplified assembly is shown in Figure 49. Computing the removal volume for individual parts is handled as described in previous section. The difference between preliminary and detail design stages are again at the level of fillets at the sharp corners. The material for the parts is assumed to be cast iron. After the range of machining energy is estimated for each part, worst-case and statistical case energy ranges are computed for the assembly. For computing the worst-case and the statistical-case energy ranges, principles from tolerance analysis are borrowed. Details of equations and explanations of these concepts can be found at Case study 2 for milling assembly energy estimates. For further details regarding the tolerancing principles for worst-case and statistical-case analysis please refer to [68, 69].

Figure 49 also presents the computational result of turning energy estimation using ACIS. Figure 49 (a) and (b) shows the original parts and assembly model respectively. Figure 49 (c) and (d) shows the results of minimum bounding cylinder computation.

Table 9 shows the calculation in detail. The second column in Table 9 lists the removal volume for each part. The third column provides the range estimate of machining energy for each part of the assembly shown in Figure 49.
From the worst-case analysis, the percentage contributions for Screw, Clout and base are 74.6, 5.2 and 24.2. The percent contribution suggests that any reduction in energy for machining the Screw will have more impact on the overall. If 10% reduction is desired in the machining energy, then the Screw, Clout and Base need to be redesigned so that the machining energy reduces by 7.46, 0.52 and 2.43 percent, respectively.

4.2.4 Design Energy Estimation for Casting Operation

In order to present the domain-independent feature of the energy estimation framework, a simple computational tool for estimating casting energy during design phase has been developed. Casting involves a lot of manual operations. During design phase, it is impossible to consider details of manual
operations; therefore, the uncertainty range exists. Melting energy is the major energy consumed during casting operations. Thus, based on heat equations [73], energy consumed during casting operations can be simply calculated as:

\[
E_{\text{Casting}} = C \times M \times \Delta T
\]

Where \( C \) stands for the specific heat capacity and its value can be obtained based on the material type. \( M \) stands for the mass of final product, and \( \Delta T \) is the maximum change of temperatures during casting operations. After inputting the specific heat capacity for different materials and the location of this computational tool into knowledge files, the framework can invoke the casting tool to calculate energy estimation. The tool is named as “DMCasting” as shown in Figure 30.

![Energy Estimation Tools for Design and Manufacturing Phase](image)

**Figure 50: Casting Energy Estimation Tool in Design Phase**

Figure 50 shows the detail of casting energy estimation tool. The list of material is parsed through XML files input by knowledge engineers. The
maximum change of temperatures and the amount to be casted are prompted for user input.

4.3 Manufacturing Phase Energy Estimation Tool Based on Manufacturing Process Plan

4.3.1 Manufacturing Process Planning

Manufacturing process planning is a systematic description of methods by which products can be manufactured from raw material form to desired form [74]. In general, design data, raw material data, manufacturing process data, equipment and tooling data form the inputs for process planning [74]. The output of process planning is the process plan. Process plan plays an important role in production management. It provides necessary information for developing new products, such as tools, machines, material types, manufacturing technologies, personal requirements, etc. This information is critical for estimating energy in manufacturing phase. As discussed in Chapter 2, manufacturing phase concentrates on the details of manufacturing parameters, these parameters can be acquired from process plan. Therefore, process plan is a critical reference for energy estimations. This section will demonstrate a manufacturing tool development for energy estimation for machining operations based on process planning. Various tools for other manufacturing operations can be developed by developers in specific domains and attached to the framework. The section will discover the developed STEP-NC [75] standard for machining first, and then use the XML file and schema provided by the standard to calculate machining energy in manufacturing phase; a case study will be given at the end.
4.3.2 STEP-NC Process Plan Structure, XML and Its Schema

STEP-NC, ISO14649 [76] is an ISO standard to structure feature-based process plans for machining process such as milling operations, turning operations, etc... The XML representation (STEP Part 28) of STEP-NC structure is shown in Figure 51.

As shown in Figure 51, the XML representation includes header elements and data elements. Header elements describe the basic information of the XML
file. In data elements, each STEP-NC process plan must start from a top level entity, called Project. The Project lists the attributes of the work plan and contains working step entities in a linear order. The Workpiece entity lists the material type, raw piece property, and base shape data. This is where the framework retrieves the material information in the manufacturing process plan.

The process plan is stored under "Its elements" entity of the work plan. This portion defines the task of machining operations to be conducted. "Its feature" entity defines the geometric features of its final shape. This is not the concentration for energy estimation in manufacturing phase, yet the "its operation" entity describes how the features should be manufactured using specific cutting tool and machining data. The tool and technology information described under "its operation" entity contains all the parameters required for estimating energy in manufacturing phase (refer to Chapter 2 for details).

- "its_tool" attribute contains the tool related information such as number of teeth, diameter, etc..
- "its_technology" attribute specifies the technology related parameters such as cutting speed, depth/width of cut, feed rate, etc..
- "its_machining_strategy" attribute indicates the strategy of machining and
- "its_machining_functions" specifies the state of various machining functions.
Various benefits of using STEP-NC process plan have been recognized [74]:

(a) Manufacturing planning time and data preparation can be significantly reduced.

(b) Complete and structured data model is provided by STEP-NC with both geometric and technological information, so that no information is lost between different stages of the processes.

(c) STEP-NC is self-documented, and it can be saved, modified and fed back to the design department for redesign.

(d) XML files can be used to store knowledge, and transfer information hence enable Web based manufacturing or e-manufacturing.

Based on the discussion above, STEP-NC XML provides an important source from which the framework can retrieve information and parameters to estimate manufacturing energy. Examples of "its_tool" and "its_technology" attributes are shown in Figure 52 and Figure 53 below. These figures also present how the two attributes look like in a STEP-NC XML file from a real manufacturing process plan. A complete STEP-NC XML file is attached at the appendix of the thesis.
<its_tool>
  <endmill its_id="">
    <its_toolbody>
      <number_of_teeth>2</number_of_teeth>
      <hand_of_cut>0</hand_of_cut>
      <coolant_through_tool>False</coolant_through_tool>
      <pilot_length>0</pilot_length>
      <dimension>
        <diameter>40</diameter>
        <helix_angle>31.96233</helix_angle>
        <radial rake>9.17043</radial rake>
        <radial relief>6.403934</radial relief>
        <radial clearance>16.70279</radial clearance>
        <cutting edge angle>1.260838</cutting edge angle>
        <end relief>6.154613</end relief>
      </dimension>
    </its_toolbody>
    <its_cutting_edge>
      <tool offset length>0</tool offset length>
      <technological data>
        <cutting angle>0</cutting angle>
      </technological data>
    </its_cutting_edge>
  </endmill>
</its_tool>

Figure 52: Example of "its_tools" Attribute in a Sample STEP-NC XML File

<its_technology>
  <cutting speed>28.35618</cutting speed>
  <feed speed>15.78295</feed speed>
  <spindle speed>226</spindle speed>
  <depth of cut>10</depth of cut>
  <passes>4</passes>
  <width of cut>40</width of cut>
  <machined length>100</machined length>
  <machined time>6.335951</machined time>
  <removal rate>19833.44</removal rate>
  <power at spindle>1.190006</power at spindle>
  <power at motor>1.487508</power at motor>
  <torque at spindle>50.2804</torque at spindle>
</its_technology>

Figure 53: Example of "its_technology" Attribute in a Sample STEP-NC XML File
4.3.3 Methods for Calculating Manufacturing Energy in Machining Operation

4.3.3.1 Machining Energy Calculation

In manufacturing process, the cutting methods in machining operations can be broadly divided into orthogonal cutting and oblique cutting. In orthogonal cutting, the tool cutting edge is set as perpendicular to the direction of movement, while in oblique cutting, the cutting tool deviates from the orthogonal plane. Turaga et al. [77] analyzed and derived equations to calculate manufacturing power in both orthogonal and oblique cutting. For orthogonal cutting, the energy can be expressed as:

$$E = C \times f \times w \times x \times V \times t;$$

(9)

Where $C$ is a material constant at a given cutting speed and rake angle [77], $f$ is the feed rate, $w$ is the width of cut, and $x$ is a constant for the material being machined [78]. The values of constant $C$ for common materials are shown in Table 10. The constant $x$ can be looked up at tables in [77, 78]. $V$ is the spindle rate, and $t$ is the machined time during manufacturing process.

<table>
<thead>
<tr>
<th>Work Material</th>
<th>Constant C under +10º Angle</th>
<th>Constant C under -10º Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free cutting carbon steel</td>
<td>1085</td>
<td>1195</td>
</tr>
<tr>
<td>Carbon steels</td>
<td>1930</td>
<td>2125</td>
</tr>
<tr>
<td>Nickel-chromium steels</td>
<td>1780</td>
<td>1960</td>
</tr>
<tr>
<td>Nickel-molybdenum and chromium-molybdenum steels</td>
<td>1780</td>
<td>1960</td>
</tr>
<tr>
<td>Chromium-vanadium steels</td>
<td>2100</td>
<td>2310</td>
</tr>
<tr>
<td>Flake graphite cast iron</td>
<td>982.5</td>
<td>1085</td>
</tr>
<tr>
<td>Nodular cast iron</td>
<td>1175</td>
<td>1295</td>
</tr>
</tbody>
</table>
For oblique cutting, the power can be calculated as:

\[
P = 2^{\frac{1+x}{2}} \cdot v \cdot s \cdot w \cdot \frac{Cf_x d^{\frac{1+x}{1+x}}}{2\pi(1+x)R^{\frac{1+x}{2}}}; \quad [77]
\]

where \(x\) is a constant for the material being machined [78], \(v\) is the spindle speed, \(s\) is the number of teeth in the cutter, \(w\) is the width of cut, \(f_t\) is feed/cutting tooth, \(C\) is the cutting force constant mentioned in Table 10, \(d\) is the depth of cut, and \(R\) is the radius of cutter.

Therefore the energy during machining is:

\[
E = P \cdot A \cdot t; \quad [77]
\]

where \(P\) is the power calculated above, \(A\) is a constant for different types of cutters and they are listed in the table in [78], and \(t\) is the machined time.

It is evident that all the parameters required to calculate manufacturing energy in machining operation can be obtained through STEP-NC XML process plan file. The two equations above generalize energy calculations during machining. The summation of machining energy and energy consumed during idle time will be the total energy estimate during manufacturing phase. Since idle time varies significantly for each work plans and individuals, it will be prompted for user input.

Most turning operations can be considered as orthogonal cutting, while milling operations can be classified into slab milling, peripheral milling, face milling, and end milling. Slab milling and peripheral milling can be recognized as orthogonal cutting; face milling and end milling can be recognized as oblique milling. The type of machining operations can be obtained in STEP-NC XML file.
under the tags of “its_operations” and “its_tools”. A sample XML file can be found at Appendix A.

4.3.3.2 Steps to Calculate Manufacturing Energy in Machining Operation

STEP-NC process plan XML file will be adopted for calculating machining energy in manufacturing phase. The steps to calculate manufacturing energy in machining operation are presented in Figure 54. First, the XML file of process plan is parsed; then the software traverses the XML tree and checks the tag of “machining_workingstep” to identify the number of working steps in process plan.

Parse XML of process plan

Traverse XML tree, check the numbers of tag: “machining_workingstep”, record the number

Based on the number recorded, loop each working step, check tag “its_operation”, decide to use orthogonal or oblique equation for energy estimation

Acquire material type for each working step in the loop

Acquire parameters needed for energy estimation in each loop

Compute machined energy consumption in each working step and in the whole process plan

Prompt user to input idle time, and compute total energy in manufacturing phase

Figure 54: Steps to Calculate Manufacturing Energy in Machining Operation
Based on the recorded number, the software starts to loop inside each working step, and checks a tag called “its_operation” to decide whether to use orthogonal or oblique equation for energy estimation. Then the software queries the material type of each working step and acquires parameters needed for energy estimation. Using these parameters, machined energy consumption in each working step and in the whole process plan are computed. At last, the tool prompts user to input idle time and compute idle time energy. The summation of idle time energy and machined energy forms the output of energy estimation in manufacturing phase for machining operations.

4.3.4 Case Study

A case of STEP-NC XML file for process plan is given in [79]. The part model is shown in Figure 55.

![Figure 55: Example Part Model Given in [79]]

The full content of the XML file is attached in Appendix A. The following figures show the tool to calculate energy estimation in manufacturing phase for machining operation.

The first scenario of using this tool is to open and load the process plan XML file as shown in Figure 56.
Figure 56: Mfg Energy Estimation Tool: Open and Load XML

After selecting XML file, the next step is to parse the process plan. Figure 57 shows the details of the parsing result for the process plan.

Figure 57: Mfg Energy Estimation Tool: Parse Process Plan XML.
Figure 58 shows the manufacturing energy estimation tool. User is prompted to input either the exact number of idle time power or the percentage of the idle power over machining power. Idle time is also prompted for user input. After collecting all the necessary parameters, machining energy and idle energy are computed and added together to generate the overall energy estimate in manufacturing phase. The computational result is popped out to inform users and stored in a separate XML file for later integration as well.

![Figure 58: Mfg Energy Estimation Tool: Energy Estimation.](image)

4.4 Use Phase Energy Estimation Tool

According to the discussion in Chapter 2, calculating energy in use phase requires parameters of product's power and time to be used. These two parameters can both be prompted for user input. Figure 59 shows the tool to calculate energy consumption during use phase.
The method for calculating energy during use phase can be simply expressed as:

\[ E_{use} = P_{use} \times T_{use} \]  \hspace{1cm} (12)

Where \( E_{use} \) is the use phase energy, \( P_{use} \) is the product's power, and \( T_{use} \) is time to be used during use phase. After calculating the use phase energy consumption, the framework will pop out a message box to provide the computational results to the user and record the results into a separate XML file for later use. The use of the separate XML file will be illustrated in Chapter 5.
4.5 Disposal Phase Energy Estimation Tool

Similarly, based on the discussion in Chapter 2, calculating energy in disposal phase requires parameters of material type, specific energy during disposal, and the amount to be disposed. Parameters of material type and the amount to be disposed are prompted for user input. Some of the specific energy during disposal phase is provided in [80]. Figure 60 shows the tool to calculate energy consumption during disposal phase.

\[ E_{disp} = SpecE_{disp} \times M_{disp}; \]  

(13)
Where $E_{dsp}$ is disposal phase energy, $SpecE_{dsp}$ is specific energy during disposal, and $M_{dsp}$ is the amount to be disposed. Similarly as in material extraction phase, the list of materials is parsed based on the XML file. The framework can parse the corresponding XML file provided by knowledge engineer, and display its list of materials dynamically. This dynamic behavior ensures the framework to meet the customizable and expandable requirements. After calculating the disposal phase energy, the framework will pop out a message box to provide the computational results to the user and record the results into a separate XML file for later use. The use of the separate XML file will be illustrated in Chapter 5.

4.6 Product Life Cycle Energy Estimation Tool and Computational Results Feedback

After conducting energy estimations in extraction phase, design and manufacturing phase, use phase, and disposal phase, computed energies are saved in separate XML files. These files are then collected and parsed by the framework to compute product life cycle energy consumption. Product life cycle energy can be simply expressed as the summation of each phase’s energy consumption. After summarizing each phase’s energy consumption, the framework creates an XML file to record the energy estimates made in previous phases and in the whole product life cycle. User can retrieve the computational data by clicking “CmptRslt” button in the frame. The framework provides computational results feedback to users by parsing the result XML file saved in last step. Figure 61 shows the tool for calculating energy consumption during the
whole product life cycle and an example regarding how these computational results feed back.

Figure 61: Energy Estimation Tool: Whole Product Life Cycle and Computational Results Feedback
CHAPTER FIVE: INTERACTION BETWEEN ENERGY ESTIMATION FRAMEWORK AND ENERGY COMPUTATIONAL TOOLS

5.1 Data Flow of Energy Estimation Framework

In order to make the framework be customizable and flexible to fulfill the requirements discussed in chapter 2. The framework has to deal with different data sets. These data sets include the XML file that knowledge engineer defines and edits, the user input data that estimation user specifies, the specific data for energy calculation in specific computational tool, and the computational results that these tools feed back. The framework is considered to be an integration of various separate pieces, thus, it can accept data input from both knowledge engineer and estimation user. It is important to make sure that the computational tools send the computed energy consumption results back to framework, so that both framework and its computational tools have access to the computational energy results. With this behavior, the framework achieves the domain independent goal and can be generic; it will also be able to either accept inputs or generate outputs. Figure 62 illustrates the data flow in the energy estimation framework. First, the framework accepts data input from both knowledge engineer and estimation user, analyzes and stores them. Then the framework generates data for specific energy computational tools according to the parameters that the tools require. The tools perform calculations and send computational results back to the framework. Finally the framework outputs the result to clients (knowledge engineer or estimation user).
Figure 62: Data Flow of Energy Estimation Framework

The above design of data flow satisfies the requirement analysis listed in chapter 2. Since the XML file is edited separately by knowledge engineer, it isolates domain knowledge from the framework. The framework is not responsible for determining which factors are required for energy calculation in specific domain; while the knowledge engineer can specify knowledge in the domains that they own. Therefore, the framework is domain independent and expandable. The tools conduct energy calculations, and send results back to framework; this ensures that the framework is an integrated piece of whole. Therefore, it is flexible to estimate energy consumption in any phases, and can accept inputs as well as output results. Estimation user can compute energy according to their customized needs.
5.2 Communication Between Energy Estimation Framework and Its Computational Tools

The communication between energy estimation framework and its computational tools is vital for integrating the framework with its attached tools. The detailed communication flow is presented in Figure 63. In the communication behavior, XML files play the role of a communicator. The data flows via XML files. They store and transfer knowledge among different components. There are four main components involved in the communication behavior. They are energy estimation users, knowledge engineers, energy estimation framework, and energy computational tools.

![Figure 63: Communication between Energy Estimation Framework and Its Computational Tools](image-url)
(1) Energy Estimation Users:

Energy estimation users take advantage of the framework to calculate energy consumptions. They are responsible for inputting product part/assembly model to the framework. They are also responsible to input parameters such as material type that have to be specified by users to perform energy estimations.

(2) Knowledge Engineers

Knowledge engineers contribute their knowledge in specific domains through XML files; these XML files form the input of the framework.

(3) Energy Estimation Framework

The framework has three main responsibilities:

(a) Parse XML file input from knowledge engineer, read product model, and read user specified information.

The framework provides two different interfaces for both knowledge engineers and estimation users. Knowledge engineers can contribute their domain knowledge into XML/Schema file. Estimation users can specify product model and input user defined information such as the materials of the product. These inputs contain information that is required to compute energy estimation for the domain. The energy estimation framework is responsible to read and parse all these information for future use.

(b) Generate knowledge that will be used for corresponding computational tools through XML.

After parsing the existing input file, the framework analyzes the data and generates a separate XML file which will be used for corresponding
computational tools. When these tools are invoked, they can parse such XML file and perform energy calculations. Computational tools are responsible for calculating energy consumptions during specific phases. Tools can read XML files that are generated from the framework, and can also produce a separate XML file which contains the computational results. The computational results will be sent back to framework.

(c) Read the computational results sent back by the computational tools and display the results to users.

After performing energy calculations, the computational results are held by computational tools in their self-generated XML files. The framework also has access to these computational results. Therefore, the framework is also responsible for reading the computational results sent back by the computational tools, and displaying the computed results to users.

(4) Energy Computational Tools

Energy computational tools also have three main responsibilities. They read the XML files generated by the framework to obtain the parameters required for energy calculations. They perform energy estimations for each phase and for the whole product life cycle. They hold computational results in their self-generated XML files and feed the results back to the framework.
CHAPTER SIX: INDUSTRIAL TEST CASES OF ENERGY ESTIMATION

FRAMEWORK

In this chapter, three industrial test cases will be conducted to validate the energy estimation framework and its computational tools. Visual representation of the computational results using pie chart will be presented for both parts and assemblies.

6.1 Industrial Test Case 1: Cardan Adapter

Figure 64 shows an industrial cardan adapter provided by Antrieb OOO Company in Russia. The product is made of steel, and the steel is imported from China. Complete energy estimation for this product will be conducted using the developed energy estimation framework.

![Figure 64: Industrial Case: Cardan Adapter](image)
i) Decide client type and input knowledge files

Prior to energy estimation, the type of client has to be decided and specific domain knowledge has to be input to the system through XML file. For this product, specific energy of materials extraction, design and manufacturing, and disposal are needed as the required parameters to calculate energy. Knowledge can be input by knowledge engineer through the knowledge management interface of the energy estimation framework.

Figure 65: Decide Client Type

Figure 66 and Figure 67 show the knowledge files (XML) and their schema (XSD). The schema file is used to validate whether the XML file matches the expected structure that has been predefined by this research.
“DMSpecE”, “ExtSpecE”, and “DspSpecE” refer to specific energy of design and manufacturing, material extraction, and disposal respectively. They can be obtained through [55], [58], and [80] respectively.
In addition to materials, locations of computational tools also have to be input or edited. These locations indicate where the executable files of computational tools lie. In this research, a tool to estimate machining energy in design and manufacturing phase is developed and named as “DMMachining”. Figure 67 shows the input of tool locations. The content of knowledge files (XML) can be added, modified, and expanded following the schema files predefined by this research.
ii) Parse XML files

After the knowledge file (XML) has been inputted, the framework can parse it for future use. Figure 68 shows the parsing result of previously inputted XML file for materials.

![Figure 68: Parse XML File](image)

iii) Switch client type and input user specified data

In order to estimate energy, the client type has to be switched to “Energy Estimation User” to enable the energy estimation user interface. User needs to specify input data, and in this case, they need to specify the material for the product as steel. These data are stored by the framework as shown in Figure 69(b) and will be sent to any tools which might require these data to calculate energy upon request. Figure 69 shows the dialog for user to specify data.
iv) Estimate extraction energy

The extraction energy estimation tool can be invoked by clicking “EE-MEx” button. The tool will ask users to specify the amount to be extracted and the material type if they haven’t specified any in previous step. Figure 70 presents the extraction phase energy estimation for the cardan adapter. The volume of the stock for this product can be either directly obtained from the manufacturing
vendor or by using the tool in design phase to compute the bounding box which can be approximately recognized as the raw stock. The material is selected as “steel”, the volume of the stock to make the product is provided as 4328.13 cubic centimeters, and its mass is 34.21 kg. After calculating extraction phase energy, a message box will be popped out, and the energy will be saved in a separate XML file.

Figure 70: (a) Extraction Phase Energy Estimation for Cardan Adapter (b) Result Stored by Extraction Phase Energy Estimation Tool
v) Estimate design and manufacturing energy

The design and manufacturing energy estimation tool can be invoked by clicking “EE-DM” button. The framework first pops up a dialog to ask user for selecting a computational tool as shown in Figure 71. After selecting the tool, an executable file for design and manufacturing energy computation in a specific manufacture domain is invoked.

![Select Computational Tool](image)

Figure 71: Select Computational Tool

The locations of the tools are stored in an XML file called “ToolLocation.xml” as shown in Figure 67. The paths of all computational tools can be input or edited before energy estimation so that the framework is able to parse this XML file, retrieve tool paths, and call relevant tools. The tool is separated from the framework because it needs specific domain knowledge to develop for particular manufacturing processes. Various tools can be developed and attached to the framework. This research only develops tools in machining
domain for design and manufacturing phase. For this case, the product can be manufactured using milling methods. Figure 72 presents the design and manufacturing phase energy estimation for the cardan adapter. Since the vendor is not willing to give out process plan for manufacturing the cardan adapter. The manufacturing phase energy is not calculated for this case. Yet manufacturing phase energy is easy to calculate by specifying the manufacturing plan’s xml file. The computational tool will conduct the calculations automatically. For details of energy calculation during manufacturing phase, please refer to §4.3. After calculating design and manufacturing energy, a message box will be popped out, and the energy will be saved in a separate XML file.

In design stage, the tool is estimating manufacturing energy in a large range. The estimation result becomes more accurate if the manufacturing plan is provided. If there’s no manufacturing plan provided, this research will adopt the average value of the estimation made in design stage as the energy consumption in design and manufacturing phase during a whole product life cycle.
vi) Estimate use phase energy

The use energy estimation tool can be invoked by clicking “EE-Use” button. The tool will ask users to specify the power needed during use phase and the product’s life span. Figure 73 presents the use phase energy estimation. Since it’s unknown where and how this product will be used, the use phase energy will be counted as 0 for current estimation. After calculating use phase energy, a message box will be popped out, and the energy will be saved in a separate XML file.
vii) Estimate disposal energy

The disposal energy estimation tool can be invoked by clicking “EE-Dsp” button. The tool will ask users to specify the amount to be disposed and the material type if they haven’t specified any in step (c). Figure 74 presents the disposal phase energy estimation for the cardan adapter. The material is selected as steel, the volume of the product is provided as 2461.18 cubic centimeters, and its mass is 19.443kg. After calculating disposal phase energy, a
message box will be popped out, and the energy will be saved in a separate XML file.

![Image](image_url)

**Figure 74:** (a) Disposal Phase Energy Estimation for Cardan Adapter (b) Result Stored by Disposal Phase Energy Estimation Tool

viii) Estimate PLC energy, pie chart representation, and computational results retrieval

The PLC energy estimation tool can be invoked by clicking “EE-PLC” button. The tool will collect previously calculated energy consumption in each
phase and generate PLC energy estimation. Figure 75 shows the computational results of PLC energy estimation.

Figure 75: PLC Energy Estimation for Cardan Adapter
After calculating PLC energy, a message box will be popped out, and the energy will be saved in a separate XML file. Figure 76 shows the computational results stored by the framework.

Figure 76: PLC Computational Results for Cardan Adapter Stored by the Framework

The PLC energy estimation tool also provides a visual representation for the distribution of energy in each phase in a pie chart. This chart will assist designers to judge which phase has more energy consumption and to redesign. Figure 75 presents that energy consumed for cardan adapter during extraction, design and manufacturing, use, and disposal phase is 4%, 43%, 0%, and 52% respectively. Design and manufacturing phase and disposal phase consumes the most energy. Therefore, reuse of the part rather than disposal is suggested. In-depth energy analysis such as sensitivity analysis and redesign is not the focus of this research and will be left for future work. The computational results can be retrieved by clicking “CmptRslt” button, the framework will again parse the XML file that has collected energy consumption in each phase, and feed back to users. Figure 77 presents the computational results feedback for the cardan adapter.
6.2 Industrial Test Case 2: Clamping Ring

Figure 78 shows an industrial clamping ring provided by Antrieb OOO Company in Russia. The product is made of aluminum and the aluminum is imported from China. The volume of the stock to make the product is provided as 78.40 cubic centimeters, and its mass is 0.210kg. The volume of the product is provided as 27.993 cubic centimeters, and its mass is 0.074kg. Complete energy estimation for this product will be conducted using the developed energy estimation framework.
Since steps to conduct energy estimation are similar with case 1. The following content will just show the brief result calculated by the energy estimation framework for the clamping ring. Since the process plan for this product is not provided, energy in design and manufacturing phase is calculated in design stage as shown in Figure 79. Computational Results of all phases for the clamping ring is shown in Figure 80.
Figure 79: Computational Results in Design and Manufacturing Phase for Clamping Ring

Figure 80: Computational Results for Clamping Ring in Each Phase
Similarly as case 1, a visual representation of the energy distribution for the clamping ring is provided. From Figure 81, energy consumed in extraction, design and manufacturing, use, and disposal phase is 33%, 33%, 0, 32% respectively. The material is selected as aluminum, and it costs more energy than the extraction of steel as calculated in case 1. Reuse of the part is also advised since the disposal will cost one third of the total energy in the product's life cycle. Details of energy analysis for redesign the part will be left for future work. The computational result is fed back to inform users as shown in Figure 82.

Figure 81: PLC Energy Distribution for Clamping Ring
Figure 82: Computational Result Feedback for Clamping Ring

6.3 Industrial Test Case 3: Sliding Mount Units-Assembly

Figure 83 shows an industrial assembly case sliding mount units provided by Fixtureworks Company in USA. The sliding mount units are used on fixture
plates to allow for both horizontal and vertical adjustments of clamps, supports, etc. The center block slides on the right block and the left block slides on the center block. The product is made of steel and the steel is imported from China.

Figure 83: (a) Assembly of Sliding Mount Unit (b) Left Block of the Unit (c) Center Block of the Unit (d) Right Block of the Unit

The volume of part (b), (c), and (d) are provided as 56.58, 65.29, and 59.99 cubic centimeters respectively, and their mass are 0.44, 0.51, and 0.47 kg respectively. The volume of the assembly is 181.86 cubic centimeters, and its mass is 1.42 kg. Complete energy estimation for this product will be conducted using the developed energy estimation framework. The volume of the stock to make the product is not provided by the company; yet as discussed before, it can be
calculated by the machining tool for design and manufacturing phase. The volume of the stock is 252.31 cubic centimeters, and its mass is 1.97kg.

![Figure 84: Computational Results for Sliding Mount Units in DM Phase](image)

Steps to conduct energy estimation are similar with case 1 and 2. The framework will loop the assembly, perform energy calculations for each part, and add each part’s computational result together. The following content will show a brief calculation summary computed by the energy estimation framework for the assembly. Since the process plan for this product is not provided, energy in design and manufacturing phase is calculated in design stage as shown in Figure 84.
Figure 85: PLC Energy Distribution for Sliding Mount Units
Similarly as case 1 and 2, a visual representation of the energy distribution for the clamping ring is provided. From Figure 85, energy consumed in extraction, design and manufacturing, use, and disposal phase is 27%, 43%, 0, 29% respectively. Details of energy analysis for redesign of the part will be left for future work. The computational result is fed back to inform users as shown in Figure 86.

![Figure 86: Computational Result Feedback for Sliding Mount Units](image)
CHAPTER SEVEN: CONCLUSIONS, DISCUSSIONS, AND FUTURE WORK

7.1 Conclusions and Research Contributions

The main objective of this research is to develop a CAD-based software framework to estimate energy consumption during a product life cycle. The dissertation illustrates the approach of developing the energy estimation framework following software development life cycle (SDLC) steps and presents all the computational tools that are attached to the framework to compute energy in each phase and whole product life cycle. This research also develops a novel pattern to allow the communications among the framework, energy computational tools, energy estimation users, and knowledge engineers through XML files. Knowledge can be stored and transferred through XML files and flows between software and users. The framework is developed to be domain independent and flexible. It allows various energy computational tools for calculating energy consumptions in different manufacturing domains and different phases during a product life cycle to be attached to the framework. The framework is therefore expandable for different manufacturing domains and customizable for users. This dissertation also conducts test cases of industrial products to test the validity and evaluate the performance of the framework. With help of this framework, knowledge engineers who exert to integrate knowledge into computer systems can interpret domain-specific knowledge and share their expertise to improve the framework. The framework also assists users who have little knowledge about energy computations to estimate energy consumptions during the design stage. Energy estimations in each phase and during whole
product life cycle can be made and energy efficiency of products can be improved by utilizing this framework. Such estimation can be used to re-design the part and assemblies for energy efficiency.

The key contributions of this research can be described as follows:

- This research constructs a software framework to estimate energy consumptions following SDLC steps. Relevant computational tools have been developed to calculate energy consumptions during material extraction phase, design and manufacturing phase, use phase, disposal phase, and during a whole product life cycle. Given a case product, energy consumed during each phase and during the whole life cycle can be generated. A visual representation for energy distributions in each phase during a life cycle can also be presented.

- This research also develops a novel pattern to allow the communications among the framework, energy computational tools, energy estimation users, and knowledge engineers through XML files. Knowledge can be stored and transferred through XML files and flows between software and users.

- The framework is customizable for energy estimation users, so that users are able to input specified information such as material types for energy calculations, and can estimate energy consumptions in either material extraction phase, design and manufacturing phase, use phase, disposal phase, and the whole product life cycle.
• The framework is domain independent, so that it does not depend on the specific domain knowledge. It is developed as a generic shell for energy estimation in all manufacturing domains. Machining and casting energy estimation tools have been developed to validate the domain independency of the framework.

• The framework is expandable for knowledge engineers and developers. Knowledge engineers are able to use their expertise in specific domains to input and edit knowledge files (XML) to provide parameters for energy calculations. Developers can implement various energy estimation tools for different manufacturing domains or for different phases of a product life cycle and attach the executable files to the framework. The framework allows the expansion of computational tools because of the novel communication pattern mentioned above. All the locations for computational tools can be saved in an XML file, and the framework is able to parse the XML file and generate a list of tools for estimation users to select.

7.2 Research Limitations

The main emphasis in this thesis has been to develop a software framework for energy efficient product life cycle. However, due to a large number of practical limitations, the current implementation has several restrictions. Some of the main limitations are described below.

• Presently, the framework is not integrated with experimental data and will not consider tolerance during design and manufacturing; the integration
and tolerancing is left for future work. Thus, estimates provided by the framework may differ from those in actual manufacturing experiments.

- Energy consumption during a product life-cycle is also related with the supply chain including location of manufacturers and distributors, transportation methods, and so on. In this research, supply chain is not considered.

- Various individual impacts such as different setups and manufacturing operations will affect estimated results. This study provides generic estimates based on the domain knowledge in typical situations. Unless detailed information for individual case is provided, the framework cannot make specific energy estimations.

- Energy consumption for disassembling is not considered in the framework. At the end of a product life cycle, only energy consumption at disposal phase is calculated.

- In this research, data for materials, specific energy for extraction phase, manufacturing phase, and disposal phase, manufacturing resources, and manufacturing parameters are far from complete. However, due to the domain-independent feature of the framework, such information can be expanded through diverse developers and knowledge engineers.

- This research only develops one computational tool in design and manufacturing phase to calculate energy for machining operations. The objective for development of this tool is to demonstrate the implementation of computational tools that require domain knowledge and to present how...
to attach various tools to the framework through executable files. However, due to the expandable feature of the framework, tools for different manufacturing operations can be implemented by developers and attached to the framework.

7.3 Recommendations for Future Work

In order to improve the energy estimation framework for real-life products, several extensions will be needed for this research. Below are some of the recommendations for future extensions.

- Experimental data for a large number of materials and multiple machining operations will need to be collected to support the framework.
- Supply chain, tolerance in design and manufacturing phase, disassembly energy, and recycling energy will be considered to enhance the framework for analyzing real-life cases.
- Computational tools for different manufacturing processes will need to be developed to improve the functionality of the energy estimation framework.
- Detailed energy analysis such as sensitivity analysis can be conducted.
- The framework can be integrated into large CAD systems or LCA software.

It is expected that the recommended future extensions will enhance the usefulness of this research, and will result in a large amount of reduction in energy consumptions.
REFERENCES


[58] Electrical and Mechanical Services Department, "Life Cycle Energy Analysis,"
[80] "Disposal calculations,”
APPENDIX

Appendix A: Sample STEP-NC XML File for Process Plan

<STEP-XML xmlc="ISO 10303-28">
  <file_schema>integrated_cnc_schema</file_schema>
  <file_description>ISO14649 file</file_description>
  <project its_id="Example - 1">
    <its_workpiece its_id="">
      <its_material>
        <material_id>Free machining carbon steel</material_id>
        <material_hardness>
          <scale>BHN</scale>
          <high_value>250</high_value>
          <low_value>200</low_value>
          <nominal>230</nominal>
        </material_hardness>
      </its_material>
      <main_workplan>
        <its_elements>
          <machining_workingstep>
            <its_id>WS131</its_id>
          </machining_workingstep>
        </its_elements>
      </main_workplan>
    </its_workpiece>
  </project>
</STEP-XML>
<placement>
  <location x="109" y="50" z="83"/>
</placement>

<slot id="SQ4">
  <end_condition>
    <first_end_condition>
      <open_slot_end_type/>
    </first_end_condition>
    <second_end_condition>
      <radiused_slot_end_type/>
    </second_end_condition>
  </end_condition>
  <course_of_travel distance="100">
    <direction x_ratio="0" y_ratio="0" z_ratio="0"/>
  </course_of_travel>
  <square_u_profile>
    <width>40</width>
  </square_u_profile>
</slot>

<its_tolerance>
  
</its_tolerance>

</its_feature>
<its_operation>
  <bottom_side_rough_milling its_id="SQ4-BSRM-R">
    <overcut_length>0</overcut_length>
    <axial_cutting_depth>0</axial_cutting_depth>
    <radial_cutting_depth>0</radial_cutting_depth>
    <allowance_side>0</allowance_side>
    <allowance_bottom>0</allowance_bottom>
    <start_point x="0" y="0" z="0"/>
  </bottom_side_rough_milling>
  <its_machining_strategy>
    <overlap>0</overlap>
    <allow_multiple_passes>False</allow_multiple_passes>
  </its_machining_strategy>

  <its_tool>
    <endmill its_id="">
      <its_toolbody>
        <number_of_teeth>2</number_of_teeth>
        <hand_of_cut>0</hand_of_cut>
        <coolant_through_tool>False</coolant_through_tool>
        <pilot_length>0</pilot_length>
        <dimension>
          <diameter>40</diameter>
          <helix_angle>31.96233</helix_angle>
        </dimension>
      </its_toolbody>
    </endmill>
  </its_tool>
</its_operation>
<radial_rake>9.17043</radial_rake>
<radial_relief>6.403934</radial_relief>
<radial_clearance>16.70279</radial_clearance>
<cutting_edge_angle>1.260838</cutting_edge_angle>
<end_relief>6.154613</end_relief>
</dimension>
</its_toolbody>
<its_cutting_edge>
<tool_offset_length>0</tool_offset_length>
<technological_data>
<cutting_angle>0</cutting_angle>
</technological_data>
</its_cutting_edge>
</endmill>
</its_tool>

<its_technology>
<cutting_speed>28.35618</cutting_speed>
<feed_speed>15.78295</feed_speed>
<spindle_speed>226</spindle_speed>
<depth_of_cut>10</depth_of_cut>
<passes>4</passes>
<width_of_cut>40</width_of_cut>

<machined_length>100</machined_length>
<machined_time>6.335951</machined_time>
<removal_rate>19833.44</removal_rate>
<power_at_spindle>1.190006</power_at_spindle>
<power_at_motor>1.487508</power_at_motor>
<torque_at_spindle>50.2804</torque_at_spindle>
</its_technology>

<its_machine_functions>
  <coolant>False</coolant>
  <mist>False</mist>
  <chip_removal>False</chip_removal>
  <oriented_spindle_stop x_ratio="0" y_ratio="0" z_ratio="0"/>
</its_machine_functions>

</bottom_side_rough_milling>
</its_operation>
</machining_workingstep>
</its_elements>
</main_workplan>
</its_workpiece>
</project>
</STEP-XML>