ABSTRACT

INTEGRATED PRODUCT AND PROCESS DEVELOPMENT METHODOLOGIES FOR ENVIRONMENTALLY CONSCIOUS ELECTRONIC PRODUCTS

by
Pingtao Yan

This research focuses on integrated product and process development (IPPD) methodologies for environmentally conscious electronic products. After a review of current research issues in the field of product and process development, a generic framework for IPPD is proposed which describes most of the concerned issues formally as constrained optimization problems. These problems may include such optimization objectives as cost, benefit, and environmental impact. Based on this framework, an IPPD methodology is proposed as a systems approach to competitive and environmentally conscious product and process development. A case study on personal computer development is performed illustrating how to apply the methodology meaningfully and efficiently. Eco-compass concept is then integrated into the methodology to evaluate environmental impact, and a case study on business telephone development is performed. To automate the design of products and processes, a solution methodology for IPPD based on logical representation of process relations is proposed with two illustrating product development examples. Finally, a timed IPPD methodology is introduced with increased modeling capability and decision accuracy. It considers the execution duration of processes and their time-varying characteristics. The timed methodology is applied to the life cycle development of flexible manufacturing systems (FMSs) and provides a new way to develop cost-effective, high-quality, and environmentally conscious FMSs.

INTEGRATED PRODUCT AND PROCESS DEVELOPMENT METHODOLOGIES FOR ENVIRONMENTALLY CONSCIOUS ELECTRONIC PRODUCTS

by
Pingtao Yan

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

January 2000

APPROVAL PAGE

INTEGRATED PRODUCT AND PROCESS DEVELOPMENT METHODOLOGIES FOR ENVIRONMENTALLY CONSCIOUS ELECTRONIC PRODUCTS

Pingtao Yan

Dr. MengChu Zhou, Dissertation Advisor Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Donald H. Sebastian, Dissertation Co-Advisor Date
Professor of Industrial and Manufacturing Engineering, NJIT

Dr. John D. Carpinelli, Committee Member Date
Associate Professor of Electrical and Computer Engineering, and
Computer and Information Science, NJIT

Dr. Reggie J. Caudill, Committee Member Date
Professor of Industrial and Manufacturing Engineering, NJIT

Dr. Edwin Hou, Committee Member Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Venkatesh Kurapati, Committee Member Date
Senior Technical Staff Member, AT&T, Middletown, NJ

BIOGRAPHICAL SKETCH

Author: Pingtao Yan

Degree: Doctor of Philosophy in Electrical Engineering

Date: January 2000

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2000
- Master of Engineering in Electrical Engineering,
  Xi'an Jiaotong University, Xi'an, P. R. China, 1994
- Bachelor of Engineering in Electrical Engineering,
  Xi'an Jiaotong University, Xi'an, P. R. China, 1992

Major: Electrical Engineering

Presentations and Publications:

P. Yan, M. Zhou, and D. Sebastian, "A generic framework for integrated product and process development," *Int. J. of Environmentally Conscious Design and Manufacturing*, vol. 7, no. 3, pp. 47-57, 1998.

P. Yan, M. Zhou, and D. Sebastian, "An integrated product and process development methodology: Concept formulation," *Robotics and Computer-Integrated Manufacturing*, vol. 15, pp. 201-210, 1999.

P. Yan, M. Zhou, B. Hu, and Z. Feng, "Modeling and control of workstation level information flow in FMS using modified Petri nets," *J. of Intelligent Manufacturing*, vol. 10, no. 6, 1999.

P. Yan, M. Zhou, D. Sebastian, and R. Caudill, "Integrating eco-compass concept into integrated product and process development," submitted to *Int. J. of Environmentally Conscious Design and Manufacturing* for review, 1999.

BIOGRAPHICAL SKETCH
(Continued)

Presentations and Publications:

P. Yan and M. Zhou, “A solution methodology for integrated product and process development based on logical representation of process relations,” submitted to *IEEE Trans. on Sys., Man, and Cyber.* for review, 1999.

P. T. Yan, Q. Z. Ye, S. H. Ong, and S. C. Tan, “A flexible snake model,” in *Proc. Int. Conf. on Control, Automation, Robotics and Vision*, Singapore, Dec. 1996, pp. 2321-2325.

P. T. Yan, Q. Z. Ye, S. H. Ong, and S. C. Tan, “Initializing snakes: A stepwise expanding flexible balloon,” in *Proc. Scandinavian Conf. on Image Analysis*, Lappeenranta, Finland, June 1997, pp. 667-674.

P. Yan, M. Zhou, B. Hu, and Z. Feng, “Modeling and control of the FMS workstation level information flow by a modified Petri net,” in *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Los Angeles, CA, Sept. 1997, pp. 321-326.

P. Yan, M. Zhou, and D. Sebastian, “Formulation of a generic framework for integrated product and process development,” in *Proc. IEEE Int. Conf. on Sys., Man, and Cyber.*, San Diego, CA, Oct. 1998, pp. 2609-2614.

P. Yan, M. Zhou, and D. Sebastian, “An integrated product and process development methodology: A case study,” in *CD-ROM Rensselaer's Int. Conf. on Agile, Intelligent, and Computer-Integrated Manufacturing*, Troy, NY, Oct. 1998, Paper No. 46.

P. Yan, M. Zhou, and D. Sebastian, “Multi-lifecycle product and process development: Selection of optimal production, usage, and recovery processes,” in *Proc. IEEE Int. Symp. on Electronics and the Environment*, Danvers, MA, May 1999, pp. 274-279.

P. Yan, M. Zhou, and R. Caudill, “A life cycle engineering approach to FMS development,” submitted to *IEEE Int. Conf. on Robotics and Automation*, Sept. 1999.

P. Yan and M. Zhou, “A solution methodology for IPPD based on logical representation of process relations,” submitted to *IEEE Int. Symp. on Electronics and the Environment*, Oct. 1999.

To my beloved family

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisor, Professor MengChu Zhou, and co-advisor, Professor Donald H. Sebastian, for their guidance, friendship, and moral support throughout this research.

Special thanks to Professors John D. Carpinelli, Reggie J. Caudill, Edwin Hou, and Dr. Venkatesh Kurapati for serving as members of the committee.

I am grateful to the support of the New Jersey Commission on Science and Technology and such industrial firms as Lucent Technologies, AT&T, Panasonic, and All Technics Products Inc. through the Multi-lifecycle Engineering Research Center (MERC) at NJIT.

Many of my fellow graduate students in the Discrete Event Systems Laboratory and the Multi-lifecycle Engineering Research Center are deserving of recognition for their support. I also wish to thank Lisa Fitton and Elizabeth McDonnell for their assistance over the past three years.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Background

A product is something sold by an enterprise to its customers. The economic success of manufacturing firms largely depends on their ability to identify the needs of customers and to create quickly products that meet these needs and can be produced at low cost. Product development is the set of activities beginning with the perception of a market opportunity and ending in the production, sale, and delivery of a product [Ulrich and Eppinger, 1995].

New product development has always been exciting, challenging, and most of all, very complex [Hainer, Kingsbury, and Gleicher, 1967; Buggie, 1981]. Throughout the history of industrial and social development, the introduction of new products has always been a close reflection of the developments in science and technology of the time. New scientific and technological developments and discoveries are transformed into new product features and better product performances to increase customer satisfaction and ultimately to bring more profit to an enterprise.

During the last two decades, science and technology are developing at an accelerating speed and widening broadness. New techniques, new materials, and new processing methods are coming up even more widely and frequently than ever. This is good and also bad news for product developers. On the one hand, these new techniques and new materials bring about many new selections and opportunities that product developers could use in their new product development processes to produce better products. On the other hand, more options inevitably mean more complexity in the related decision making processes. Consequently, product development is becoming more complex and therefore more difficult along with scientific and technological development.

In addition to scientific and technological development, social and economic evolution also brings about more complexity and difficulty to a product development process. Customers are becoming much more demanding. Market competition is more intense than ever. New requirements for products arise from many new aspects: the society, government, environment, and so on. Companies under great pressure are making every effort to produce better products faster than their competitors, otherwise, their profit would not be attainable.

Among the social and economic issues concerning a product, environmental concerns are receiving more and more attention in the recent years. The demands to create eco-efficient products without compromising their cost, quality, and schedule constraints are increasing. An eco-efficient product may be defined as a product that both minimizes adverse environmental impact and maximizes conservation of valuable resources throughout its life cycle. Environmental impact includes energy consumption, waste emissions, health and safety impact, and so on.

Several factors have contributed to the growing interest of manufacturing firms in eco-efficiency [Fiksel, 1996]:

(1) Market awareness: Government, industrial, and retail customers are all increasingly conscious of the environmental performance of suppliers and products;

(2) Differentiation: Eco-efficient designs are generally superior in terms of elegance, energy conservation, and cost of ownership, and may sway a purchase decision if price and performance are comparable;

(3) Cost savings: Eco-efficient products and processes can make a significant contribution to product line profitability through savings in production, distribution, and other life cycle costs;

(4) Eco-labeling programs: A number of product eco-labeling initiatives have arisen both in the U.S. and abroad, and the European Union is moving towards an eco-labeling standard;

(5) Regulatory pressures: Regulations governing the environmental impact of products and production processes are becoming more stringent worldwide, especially with regards to waste disposal and recycling. Japan and several countries in Europe are enforcing product take-back laws governing automobile and electronic products; and

(6) International standards: The International Organization for Standardization (ISO) has established Technical Committee TC207 to develop a global consensus on environmental management standards.

Perhaps the most important factor in changing industry attitudes is the realization that paying attention to environmental responsibility can actually increase profitability. Reducing pollution at the source and designing products and processes in ways that enhance environmental quality generally result in higher productivity and reduced operating costs, and may also increase market share.

In response to growing levels of environmental concern, many companies saw the social and economic value of shifting their attention from cleaning up waste, to cleaning production processes, and finally to eco-design of products.

In Europe and the U.S., where environmental degradation has been so marked and community awareness so advanced, companies have recognized that their future viability depends on their competitive response to these changing circumstances. In global terms, it is the extent of the redesign of manufactured products to improve their environmental performance that best illustrates the impact of global concern for sustainable development.

There has been enough activity in the redesign of manufactured goods around the world to discern some clear stages of development. Initial eco-re-designed products simply reflected market pressures and concentrated on those elements that were the easiest to change. Concern with materials and solid waste, energy consumption, and the elimination of obvious toxic chemicals combined with significant attention to packaging defined the "first wave" of such products. Their environmental contribution was minimal, but they served to heighten and sharpen market demand.

Later eco-designed products began to reflect increased attention to some full life cycle appreciation of environmental impact, with greater focus on recycled materials, design for disassembly, recyclability, and other strategies addressing problems resulting from production, use, and disposal.

Among industrial products, consumer electronics deserve substantial attention because of their tremendous environmental impact. A typical desktop computer contains many reusable and/or hazardous materials including lead, silver, gold, cadmium, plastic, and glass. When a personal computer is disposed in a landfill or incinerator, not only is the value of those materials lost, but the environmental costs of disposing of this equipment can be high as well. It has been estimated that 70% of the toxins in landfills are accounted for by 1% of the landfills' contents, principally consumer electronics.

The scope of this problem is enormous. More than 17 million computer displays were sold in 1994. It has been estimated that by the end of 1995, over 300 million televisions or computer monitors were in use. The Electronics Industry Association estimates that between 1991 and 1993, sales of televisions increased by more than 18%, sales of telephones increased by 20%, and sales of personal computers rose by almost 22%. Over 2 million pounds of plastic is used each year to manufacture these consumer electronics. However, less than 3% of that plastic is recycled.

New product development has never been easy, and it is becoming more difficult than ever before. A great amount of research effort is being pursued worldwide, trying from different perspectives to solve this problem.

1.2 Motivation

As shown in Section 1.1, product developers have a whole host of goals to optimize as they design their products: high performance, high reliability, low cost, attractive appearance, safety, and carefully chosen environmental impact.

The following environmental interactions are among those that the developers need to consider as they develop their products and processes [Socolow *et al.*, 1994; Graedel and Allenby, 1995]:

(1) Choosing raw materials: In the long term, society cannot be sustainable if it uses up important resources that are in short supply, e.g., copper and petroleum. Industrial product and process designs using any of the potentially limited materials should, if possible, be avoided;

(2) Minimizing and specifying air emissions: Most industrial processes involve the emission of materials to the air. The most common emittants are solvents or cleaning agents, which are usually organic materials, often halogenated. Product developers should attempt to avoid processes that involve the emission of CFCs, halons, CH_4, N_2O, NO_x, or volatile organic carbon (VOC). Alternatively, these materials may be captured prior to their emission;

(3) Minimizing and specifying liquid waste: Industrial operations use water for transport, cooling, and processing. In addition, organic solvents and other spent liquids constitute a portion of the liquid waste that industry must process or discard. The minimization of liquid waste, and the design of it to make it less costly to recycle than to discard, should be a central element in process design;

(4) Minimizing and specifying solid waste: The disposal of solid waste has become one of the major problems of developed societies. In Europe, landfill capacity is so exhausted that only several years' dumping at the present rate is possible. In the U.S., about two-thirds of all landfills have closed within the past decade. Solid waste disposal has thus become increasingly expensive. Industry, which generates a substantial fraction of all solid waste, is increasingly under pressure to minimize its rate of solid waste production and disposal;

(5) Designing for energy efficiency: Manufacturing activities consume 25-30% of all energy use in the U.S. and similar fractions elsewhere. Much of this consumption can be minimized by attention to process design and by the reuse of energy expended by manufacturing processes. Another aspect of design for energy efficiency is the minimization of energy consumption by products once they are in service; and

(6) Recycling after use: The materials cycle in industrial ecology is effective only if materials are efficiently returned to the system for reuse. This process is greatly aided by product designs that aid disassembly, avoid the use of a multiplicity of materials or of irreversible materials bonding techniques, identify the materials used in the product, and avoid or minimize the inclusion of materials difficult or dangerous to recycle.

As an example, let us consider the development of a cup. Traditionally, requirements for a cup come from the user. The user requires that a cup be aesthetically pleasant, microwavable, able to endure temperature variations without breaking, etc. Recently, however, more and more requirements arise from people other than the user. For example, the manufacturer requires that a cup be easy to produce, production cost be as low as possible, etc. After a cup is used and disposed of, the recycling industry requires that recycling be easy, recycling cost be low, and hazardous emissions be minimum, etc. These different requirements from different life phases of a product merge into the need for an overall consideration of a product's entire life cycle and ultimately an integrated product and process development environment.

A great amount of research effort has been pursued worldwide to achieve these goals, among which are two critically important areas, namely, integrated product and process development and design for environment. Integrated product and process development (IPPD), also called life cycle engineering or concurrent engineering, is a systematic approach to the integrated, concurrent design of products and their related processes. It is intended to cause the developers, from the outset, to consider all the elements of a product's life cycle from conception through disposal, including quality, cost, schedule, and user requirements. Design for environment (DFE) is a systematic consideration of design performance with respect to environmental, health, and safety objectives over the full product and process life cycle. It takes place early in a product's design phase to ensure that the environmental consequences of a product's life cycle are understood before manufacturing decisions are committed.

Integrated product and process development methodologies for environmentally conscious electronic products are the focus of this research.

1.3 Objective

The goal of this research is to develop systematic and formal approaches to competitive and environmentally conscious product and process development. Specific objectives include:

(1) Provide formal and mathematical models for generally considered product and process development problems;

(2) Provide systematic, formal, and efficient methods to solve these problems;

(3) Integrate fully environmental considerations into product and process development; and

(4) Perform case studies on electronic product development using the proposed concepts and methods.

1.4 Organization

This dissertation is organized as follows. Chapter 2 reviews the current research issues in the field of product and process development. Chapter 3 proposes a generic framework for IPPD, by which most of the concerned issues are formally described as constrained optimization problems. Based on this generic framework, an IPPD methodology is introduced in Chapter 4. Important concepts of the methodology are defined and an application procedure is provided illustrating how to apply the methodology systematically and efficiently to real product and process development. Chapter 5 provides a case study on personal computer development using the proposed methodology.

In order to enrich the methodology with respect to concrete evaluation of environmental impact, Chapter 6 integrates eco-compass concept into the methodology and provides a case study on business telephone development. Chapter 7 proposes a solution methodology for IPPD based on logical representation of process relations to automate the design of products and processes. Chapter 8 introduces time variable into the IPPD methodology to increase its modeling capability and decision accuracy, and applies it to the life cycle development of flexible manufacturing systems. Finally, Chapter 9 gives the conclusions and some future research directions.

CHAPTER 2

LITERATURE REVIEW

As mentioned in Chapter 1, new product development is becoming more complex and challenging along with scientific/technological development and social/economic evolution. Significant research effort is being carried out in industries and academia addressing various problems in this area. The current research issues generally fall into two categories: product profitability issues and environmental consciousness issues. This chapter reviews the major issues in these two categories.

2.1 Product Profitability Issues

From the perspective of the investors in a for-profit enterprise, successful product development results in products that can be produced and sold profitably. Product profitability has always been the most important goal driving product development. Since profitability is usually difficult to assess directly, four more specific dimensions, all of which ultimately relate to profit, are commonly used to assess the performance of a product development effort. These four dimensions are: product quality, product cost, development time, and development cost [Ulrich and Eppinger, 1995]. High performance along these four dimensions should ultimately lead to economic success.

Research issues in this category focus on improving the profitability of a product through better development processes. These issues are typically represented by the concept of concurrent engineering, or similarly known as simultaneous engineering, life cycle engineering, or integrated product and process development (IPPD) [Ishii, 1990;

Allen, 1990; Syan and Menon, 1994]. Concurrent engineering evolved as an alternative to the conventional sequential engineering method of product development. In a sequential engineering approach, each development stage starts only when the previous one is completed, which brings about major shortcomings that reduce the desired profitability [Hunt, 1993; Syan and Menon, 1994].

In its original definition, concurrent engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support [Syan and Menon, 1994]. This approach is intended to cause the developers, from the outset, to consider all the elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements [Syan and Menon, 1994; Yoshimura, 1994].

[Kalyan-Seshu and Bras, 1998] focuses on enabling the quantification and enhancement of product assemblability, serviceability, remanufacturability, recyclability, demanufacturability, and life cycle impact during product design. [Stuart, 1998] discusses materials selection for life cycle design. Research presented in [Chen and Liu, 1999] aims to provide on-line cost evaluation and advisory to help product designers avoid cost-ineffective design. [Hatch and Badinelli, 1999] describes a methodology for making the decisions associated with the concurrent engineering of a product and its downstream field support. [Huang and Mak, 1998] proposes a dynamic transformation approach from a sequential engineering environment to a concurrent engineering environment by combining the focused application of "design for X" with the extensive use of business process re-engineering. [Pham and Dimov, 1998] presents a new approach to concurrent engineering that focuses on simultaneous product design and process planning. The proposed approach provides a natural way for conveying manufacturing information to the designers and applies a wide range of artificial intelligence techniques for knowledge acquisition and deductive reasoning. [Culbreth, Miller, and O'Grady, 1996] presents an approach to concurrent engineering systems for flexible automation that uses constraint networks to represent the constraining influences of the automation process. [Soyucayli and Otto, 1998] describes a means of exploring design and manufactured quality as a systems level concurrent engineering support tool.

[Yassine, Chelst, and Falkenburg, 1999] uses risk and decision analysis methodologies to determine the best use of sequential, partial overlapping, and concurrent engineering execution strategies. [Eldin, 1997] assesses the use of concurrent engineering as a schedule reduction tool using four case studies to evaluate the potential for concurrent engineering to reduce project delivery times without increasing project overall cost. The study carried out in [Poolton and Barclay, 1996] shows that there are many benefits to introducing concurrent engineering and that firms differ with respect to their needs for the concurrent engineering approach.

Factors of successful concurrent engineering implementations are also discussed. A theoretical and interdisciplinary model that considers the critical human and organizational variables for success in concurrent engineering is presented in [Duffy and Salvendy, 1999]. This model aims to determine the impact of organizational ergonomics on work effectiveness in concurrent engineering. [Litsikas, 1997] discusses the importance of management support and communication in concurrent engineering. [Tom, 1997] points out that true concurrent engineering cannot happen until the entire supply chain is fully, electronically integrated. Logistics involvement in the early phases of product design and development in a concurrent engineering environment is also found to be important [Dowlatshahi, 1996].

Concurrent engineering has wide applications in many fields [Shina, 1993]. Major U.S. manufacturers such as Ford Motor Co. and Xerox Co. have successfully implemented concurrent engineering to improve time-to-market and product quality [Blau, 1994]. Dutch company Philips Electronics credits concurrent engineering with helping it to develop a color LCD in record time while at the same time dramatically reducing production costs. The concurrent engineering approach has also helped to cut development time at German machine tool builder Hermie [Blau, 1994]. A number of electronic design automation vendors have introduced design tools that cater for more concurrent engineering by either focusing on the package itself or by offering a complete suite of tools that allows each stage of the design to communicate easily using the same language [Mannion, 1998]. Caterpillar Inc.'s Track-Type Tractor division carried out a company-wide concurrent engineering initiative and yielded increased flexibility and shortened delivery times for the production of bogies [Bates, 1997]. [Meller and Gau, 1996] reports a trend toward concurrent engineering approaches to layout and production system design. [de Graaf and Kornelius, 1996] describes an attempt to introduce inter-organizational concurrent practices in a major printed circuit board manufacturing plant.

Providing concurrent engineering tools is another focus of research and development. A total of 17 European companies, including Daimler-Benz, have initiated a 6-year program to develop standard computer-aided tools and databases needed for concurrent engineering teams [Blau, 1994]. COMPASS (Computer Oriented Materials, Processes, and Apparatus Selection System) is a tool that helps design engineers identify potential manufacturing problems in the early stages of product development [Chan, King, and Wright, 1998]. It has a basic framework to provide essential information regarding production cost, cycle time, and product quality for all of the candidate processes. SPAW [Yetukuri, Yetukuri, and Fischer, 1996] is a design tool for planning a manufacturing process in a concurrent engineering environment, which is also designed to be integrated directly into a design-oriented concurrent engineering environment.

2.1.1 Special Issues in Concurrent Engineering

Special issues in concurrent engineering include quality function deployment, design for manufacture, design for assembly, and rapid prototyping [Syan and Menon, 1994].

(a) Quality Function Deployment (QFD)

QFD is a structured planning method of concurrent engineering that is used to influence the incorporation of product attributes that are in accord with customer expectations. This is done by mapping the customer requirements into specific design features and eventually into manufacturing processes. QFD is used as a systematic method to both identify and prioritize customer requirements, and to translate these requirements into product and process specifications [Syan and Menon, 1994].

A review of QFD and related deployment techniques is provided in [Prasad, 1998]. QFD is used by some firms to convert customer requirements into appropriate technical requirements. It provides a framework for determining tradeoffs among different combinations of design features. The output of the QFD gives direct input into product definition documents, customer requirements, and product specifications [Gautschi, 1993].

QFD is a very powerful tool when used by a cross-functional team in a concurrent product/manufacturing process development environment [Hales, 1994]. It uses lists and matrices to identify key information about the product or development process. Decision information is collected and weighed against customer requirements to ensure that the new product or service maintains its customer focus. Although QFD does not deal with the management aspects of developing a product, it generally improves the team's ability to define and design a product's attributes. [Goldense, 1993] gives advice on the benefits and cost of applying QFD within a company.

[Havener, 1993] discusses the way in which QFD can be effectively used to improve customer satisfaction and indicates eight steps in the QFD process:

(1) Identify customer needs;

(2) Identify product requirements;

(3) Relate customer needs to product requirements;

(4) Conduct a competitive performance assessment;

(5) Conduct a technical feasibility assessment;

(6) Perform a competitive technical assessment;

(7) Reconcile the differences between technological limitations and customers' needs by product requirement tradeoffs; and

(8) Develop important controls.

QFD has been successfully applied to a variety of industries such as automobiles, aerospace, electronics, textiles, and computer software. [Natarajan, Martz, and Kurosaka, 1999] applies QFD techniques to the design of an internal development system. The use of QFD technique as a tool for requirement acquisition and design analysis of a ground software intensive project is studied in [Elboushi and Sherif, 1997]. The results obtained indicate that the QFD process enables requirements to be captured and specifications and designs to be produced that are efficient, robust, and consistent. Telrad Telecommunication and Electronic Industries has developed a rigorous questionnaire methodology using QFD [Glushkovsky, Florescu, and Hershkovits, 1995]. Florida Power and Light (FPL), through one of its groups, launched a major customer research effort to determine what customers expect when they telephone FPL for a transaction. The group used QFD to systematically translate customer requirements into operational requirements [Graessel and Zeidler, 1993]. The use of QFD as a tool to improve software quality is examined in [Erikkson and McFadden, 1993].

QFD tools are also developed. International TechneGroup provides tools and technologies to support QFD. Its software tool, QFD/Capture, automates the compilation and prioritization of QFD data, performing all calculations and assisting in data analysis [Farrell, 1994]. In addition, the software generates the matrices fundamental to the QFD approach and ensures that customer requirements are not only built into the product, but also into the product's manufacturing process.

(b) Design for Manufacture (DFM)

Many industrial manufacturing problems and inefficiencies can be traced back to the design stage. Substantial reductions in manufacturing costs could result from revisions at the design stage and such measures would crucially affect the success of a product. DFM represents new awareness of the importance of design as the first manufacturing step. It recognizes that a company cannot meet quality and cost objectives with isolated design and manufacturing engineering operations. The objectives of the DFM approach are to identify product concepts that are inherently easy to manufacture, to focus on component designs for ease of manufacture and assembly, and to integrate both product design and manufacturing process design to make sure that the best match for needs and requirements is obtained [Syan and Menon, 1994].

[Ong and Nee, 1998] proposes a fuzzy set based fixturability evaluation procedure for establishing fixturing relations between the features on a part and assessing the suitability of these features for use as fixturing features during the machining of other features on the part. The optimization of device design for manufacture without sacrificing performance and reliability goals is discussed in [Lu, Holton, and Fenner, 1998]. The authors describe an optimized DFM procedure that uses new design of experiments, weighted least squares modeling, and multiple-objective mean-variance optimization methods. [Philpott, Warrington, and Branstad, 1996] presents a parametric contract modeler for providing manufacturing cost information to engineering designers during the early conceptual design phase of new product development. A computationally efficient method is presented in [Gadh and Prinz, 1995] to convert the basic topological elements used in CAD systems into more abstract geometrical features necessary for DFM analysis. [Bavishi, 1997] discusses the benefits of using cost-estimating software as a DFM tool.

The DFM philosophy is widely accepted in industry. [Dominach, 1993; Langan, 1998] discusses the DFM concept and application in the production of printed circuit boards and printed wiring boards. General Motors reported that it can reduce component parts and assembly time almost by half by integrating DFM schemes in its product development process [Murray, 1995; King, 1996; Bonenberger, 1994]. Ranor Inc. used a PC-based CAD/CAM system to handle the design for assembly, the design for manufacture, and the machining processes of large manufacturing tasks [Raymond, 1995]. DFM techniques were also used in the design of a new front corner chassis assembly in the fourth generation 1993 Camaro/Firebird cars [Green and Reder, 1993].

Many companies are also developing design software that utilizes the principles of DFM. These companies include Boothroyd Dewhurst, Hewlett-Packard, Matra Datavision, and Lucas [Dvorak, 1994].

(c) Design for Assembly (DFA)

It is acknowledged that assembly represents one of the major factors affecting product cost and quality. DFA is a key element in creating competitive products and reducing time-to-market. The objectives of the DFA approach are to identify product concepts that are inherently easy to assemble and to favor product and component designs that are easy to grip, feed, join and assemble by manual or automatic means [Syan and Menon, 1994].

[Tatikonda, 1994] describes a DFA methodology for product reengineering and new product development. Five factors to help companies optimize and successfully implement DFA strategies are proposed in [Munro, 1995]:

(1) Leadership has acquired knowledge of tools and experience of project teams;
(2) Leadership has developed and implemented a strategy;
(3) The plan is understood and accepted by all involved;

(4) External organizations are used for profound knowledge; and

(5) Strategic managers utilize the correct approach.

A variety of DFA tools have been developed. Boothroyd Dewhurst has released a series of versions of its programs for designing easy-to-assemble, environmentally friendly products [Deitz, 1998]. These include the Design for Assembly (DFA) software package, which guides design engineers through an analysis of a proposed design's ease of assembly and reduces part count and assembly time, the Design for Manufacture (DFM) software package, which estimates tool costs through informed material trade-off judgments, and the Design for Environment software package, which works in tandem with DFA analyses to help designers evaluate and optimize the disassembly sequences of products for end-of-life recovery. These software tools are used by automotive industry suppliers to improve product cost and quality and to gain a competitive edge over other suppliers. Lucas Engineering and Systems has also developed design-support software that is faster, easier to learn, and more integrated than Boothroyd Dewhurst's popular tools [Kobe, 1994]. The Design for Assembly part of the Lucas system identifies poor designs, difficulty of assembly, unnecessary parts, and more.

(d) Rapid Prototyping (RP)

Prototype may be defined as an approximation of the product along one or more dimensions of interest [Ulrich and Eppinger, 1995]. Prototyping is the process of developing such an approximation of the product. The main purpose of building a prototype is to help people understand the system when they define the requirements, so that they can write the requirements down in product specifications. Prototyping is aimed at reducing design time by giving an accurate specification of the problem at the beginning, which therefore removes the need for later changes [Syan and Menon, 1994].

Rapid prototyping (RP) is also referred to as solid freeform fabrication (SFF), desktop manufacturing, and layered manufacturing. These techniques have the potential to produce accurate, structurally sound, 3-D models of objects designed using computers and manufactured directly using a CAD database, from a range of materials such as photocurable resin, ceramic and metallic powders, paper, and nanophase [Manthiram, Bourell, and Marcus, 1993; Onuh and Hon, 1998]. No part-specific tooling or human intervention is required, and the models are available to the user in minutes or hours [Crawford and Beaman, 1999; Ormond, 1993].

SFF methods include stereolithography, selective laser sintering, solid ground curing, 3-D printing, laminated object manufacturing, fused deposition modeling, and recursive mask and deposit processing [Marcus and Bourell, 1993; Amon, Beuth, and Weiss, 1998; Bandyopadhyay, Panda, and Janas, 1997; Agarwala, Bandyopadhyay, and van Weeren, 1996].

The general application areas of SFF are short-run manufacturing, mold and die making, and accurate tooling [Marcus and Bourell, 1993; Sriraman, Winek, and Habingreither, 1999]. [Langdon, 1999; Chalmers, 1998] reviews recent developments in rapid prototyping and considers their likely impact on the automotive industry.

The benefits attributed to RP include cost savings, time savings, and improved product quality. However, the real value of using RP technology in product development is to reduce risk. To date, at least 99% of RP use has fitted into one of the following six applications: concept models, presentation models, functional parts, tooling patterns, investment casting patterns, and alternative tooling [Mueller, 1999]. Companies that most effectively use RP in their product development effort use a combination of these applications and utilize them at appropriate points in the development process.

2.2 Environmental Consciousness Issues

As mentioned in Chapter 1, with the rapid development of the industrial society, environmental issues are receiving more and more attention in the recent years. Demands for environmentally conscious products are a natural result of this trend. A great amount of research effort is being carried out concerning this problem [Brinkley, Kirby, and Charron, 1997; Sage, 1997; Srinivasan, Sheng and Wu, 1995a, 1995b; Ishii, Eubanks, and Di Marco, 1994; Jovane *et al.*, 1993; Navin-Chandra, 1993; Zhang and Yu, 1997; Zussman and Zhou, 1999]. In the literature, two major research areas are emphasized: life cycle analysis/assessment and design for environment.

2.2.1 Life Cycle Analysis/Assessment (LCA)

Life cycle analysis/assessment (LCA) is a family of methods for systematically assessing material and energy use, waste emissions, services, processes, and technologies associated with a product over its entire life. Three types of LCA methods are in use: life cycle inventory analysis, impact analysis, and cost analysis [Mizuki *et al.*, 1997]. Life cycle inventory analysis uses quantitative data to establish the levels and types of energy and material inputs to a system and the resulting release. Impact analysis generally involves normalization and weighting of the inventory results to formulate an overall metric for comparing dissimilar environmental results. Cost analysis derives cost properties associated with unit operations.

A comprehensive overview of life cycle assessment methodologies is presented in [De Langhe, Criel, and Ceuterick, 1998].

[Munoz and Sheng, 1995] presents an analytical approach for determining the environmental impact of machining processes. [Luo *et al.*, 1999] presents a method for analyzing the environmental performance of solid freeform fabrication processes. A technique for producing a rapid life cycle assessment is described in [Graedel, Allenby, and Comrie, 1995]. [Hendrickson, Horvath, and Joshi, 1998] shows that process and product models used for performing LCAs can be represented as process flow diagrams or as matrices of process interactions. [Saur, Hesselbach, and Eyerer, 1997] discusses enhancements to LCA that are designed to improve the quality of conclusions used as design parameters. A framework for environmental LCA based on physical measures is presented in [Knoepfel, 1996] and applied to the comparison of long-distance energy transport systems. A mixed LCA approach is used in [Schuckert, Saur, and Florin, 1996] to evaluate the environmental behavior of products or systems through an investigation of energy uses in all life stages of an industrial product from the extraction of raw materials to the disposal of wastes. An LCA approach to manufacturing process and materials selection is described in [Harsch, Schuckert, and Eyerer, 1996]. [Zhou *et al.*, 1996; Caudill *et al.*, 1997; Al-Okush, Caudill, and Thomas, 1999] introduces a multi-lifecycle concept and proposes a multi-lifecycle assessment (MLCA) methodology to evaluate the energy consumption and environmental emissions of a product. The methodology is implemented into a software package [Jin, 1999] that provides a useful decision support tool to develop products and processes that are suitable for multiple life cycles.

[Alting, Hauschild, and Wenzel, 1998] uses three industrial cases to show how a newly developed LCA methodology can assist product developers in the development of more environmentally friendly products. A study was commissioned by the U.K. Department of the Environment to assess how LCA methodology could be developed and applied to assist decision makers in waste management [Barton, Dalley, and Patel, 1996]. [Curran, 1995] describes how an LCA approach can be a beneficial support tool in assessing pollution prevention opportunities. The benefits of using economy input-output life-cycle analysis to estimate economy-wide discharges are discussed in [Lave, Cobas Flores, and Hendrickson, 1995]. The use of LCA to identify the environmental impact of products and to evaluate opportunities to reduce this impact is discussed in [Curran, 1993].

LCA has a wide range of applications. Electronic equipment manufacturer Nortel has applied LCA extensively in its overall environmental management strategy [Azapagic and Solberg Johansen, 1998]. A modified life cycle analysis (MLCA) was developed by Ford Motor Co. for the automotive industry [Sullivan, Costic, and Han, 1998]. The steel industry is gaining valuable experience in the use of LCA [Chubbs and Steiner, 1998]. [Steele and Allen, 1998] uses an abridged life cycle assessment, which produces easily comprehended information about each life stage of a product, to analyze environmental impact associated with recycling and waste management of four battery technologies. [Hazel, 1996] discusses the use of life cycle assessment as a method for evaluating the impact of coating formulation strategies on environmental and cost performance.

2.2.2 Design for Environment (DFE)

Design for environment (DFE) is a newly formed academic and technological area as people have realized that it is the product design stage that plays the most important role in the product's entire life cycle, and improvements in design bring about the greatest benefits and provide the most efficient solutions to the difficult environmental problems that may be faced later. It takes place early in a product's design phase to ensure that the environmental consequences of a product's life cycle are understood before manufacturing decisions are committed. The scope of DFE encompasses many disciplines, including environmental risk management, product safety, occupational health and safety, pollution prevention, ecology and resource conservation, accident prevention, and waste management. It is actually a combination of several design topics including disassembly, recovery, recyclability, regulatory compliance, disposition, health and safety impact, and hazardous material minimization [Mizuki *et al.*, 1997].

[Glantschnig, 1994] explores issues and challenges faced by product designers and environmental design specialists developing green products and investigates external factors and forces that impact green design. [Hersh, 1998] presents an excellent survey of systems approaches to green design and discusses their applications to the computer industry. A comprehensive overview of design for environment methodologies is presented in [De Langhe, Criel, and Ceuterick, 1998]. [Zhang, Kuo, and Lu, 1997] updates information about environmentally conscious design and manufacturing and provides some general information, guidelines, and references for research and implementation. A review of DFE approaches to sustainable development is presented in [Keoleian and Menerey, 1994]. [Sheng and Hertwich, 1998] presents an overview of the hierarchical levels of comparative waste assessment which links process-level emissions to immediate, site-wide, and eco-system impact.

DFE has many successful industrial applications. Some of the best-known environmentally progressive companies include AT&T, IBM, Hewlett-Packard, MCC, Dow Chemical, and so on [Fiksel, 1996]. In the semiconductor industry, addressing environmental issues has resulted in a cultural change in which environmental impact must be considered during product and process design [Beu and Mendicino, 1997]. The U.S. government has launched a nationwide program that aims to encourage engineers to develop more environmentally conscious designs. The National Environmental Technology Strategy calls for the promotion of innovation, the reduction of red tape, the reinvention of regulations, and improved availability of information [Wingo, 1995]. Several examples of green design are described in [Considine, 1995].

With the development of DFE theory, sophisticated design support tools are necessary for the early stage of design. These tools are able to provide guidance on design regarding to environmental impact, thus assisting designers in integrating environmental consciousness into functional considerations [Zhang and Yu, 1997]. [Thomson, Koshland, and Lucas, 1997] proposes a comprehensive environmental decision-making tool, called the index ITOX, for evaluating the effectiveness of hazardous waste destruction technologies, especially thermal-processing methods such as incineration. The Boeing Company is currently campaigning to persuade other major manufacturers to join it in creating a general purpose pollution prevention design software package. The purpose of the software, which is called the Expert Process Advisory System (EPAS), is to instantaneously present designers with information about the environmental ramifications of their design options. EPAS allows all the relevant facts to be ranked and weighted, thus improving product designs and reducing environmental life cycle costs [Betts, 1998].

2.3 Summary

The economic success of manufacturing firms largely depends on their ability to identify the needs of customers and to create quickly products that meet these needs and can be produced at low cost. New product development is becoming easier but at the same time more complex and challenging along with scientific/technological developments and social/economic evolutions.

This chapter provides an overview of the major research issues/effort being carried out in academia and industries about future products and processes. These issues can generally be classified into two categories: the category of product profitability issues and the category of environmental consciousness issues. Product profitability issues address how to develop a profitable product in more efficient ways, and environmental consciousness issues address how to minimize the environmental impact of industrial products throughout their life cycles.

This research addresses issues in both categories and provides an integrated solution methodology to competitive and environmentally conscious product and process development.

CHAPTER 3

A GENERIC FRAMEWORK FOR IPPD

Chapter 2 illustrates the diversity of research issues that are being addressed by industries and academia in the area of product and process development. Some methods and techniques are proposed and some tools are developed that are efficient and successful in solving some specific problems. However, there are no *general* methods existing that can systematically and efficiently solve the *general* problems in this area. Consequently, the overall status of new product development is still more or less ad hoc and disappointing. Successful solution of one problem does not mean much for others. When new problems arise, people do not know how to solve them as rapidly and efficiently as having been done on former ones.

However, if we look into the nature of current research issues of concern, we could notice that there exists great similarity under their seeming diversity. That is, most of these research issues have the same basic nature: they are concerned with some kind of requirements exerted on some specific phases (or the entire) of a product's expected life cycle. By this nature, most of the major research effort is trying to find some good or optimal solutions to fulfil the corresponding requirements.

Based on this observation, we then have the idea to re-address these seemingly different research issues/problems in a unified way such that we can find some *general* solutions to the *general* problems (both existing and forthcoming). As a result, a generic framework for integrated product and process development is developed in this chapter as the first major achievement of our research effort.

This chapter is organized as follows. Section 3.1 states the ideas underlying the generic framework. Section 3.2 gives the mathematical description of the generic framework. Section 3.3 explains the physical meaning of the generic framework through a simple product example.

3.1 Ideas

In the beginning of this chapter, we investigate the nature of current research issues in the area of product and process development. It is noticed that most of these research issues are dealing with some kind of requirements exerted on some specific phases (or the entire) of a product's life cycle. People are trying to find better and more systematic ways to fulfil these requirements, e.g., to reduce product development time and cost, to reduce manufacturing cost, to minimize environmental impact, and ultimately to obtain a successful product.

Based on this observation, our research is focused on looking for a methodology by which most of these seemingly different issues/problems are described uniformly and then solved similarly and formally.

To do this, a product's entire life cycle should be considered. This is because requirements for a product may arise from every step of its life cycle. In addition, a product's life cycle should be further divided into life phases. This is because requirements for a product are usually more precisely specified for each life phase than for the entire life cycle. Each life phase has its component activities and the corresponding quality of performance. Moreover, there exists close interrelationship among the product's life phases. The activities in each life phase may influence the activities in its posterior life phases. Therefore, in order to improve the performance of a life phase, not only should some requirements be exerted on this specific life phase, some other related requirements should also be exerted on its prior life phases that have influence on this one. Following this manner, most of the research issues of concern could be combined into a large constrained optimization problem and then be solved formally.

The generic framework for integrated product and process development, which will be described next, represents the ideas hereinabove.

3.2 A Generic Framework for IPPD

The mathematical description of the generic framework is as follows:

Represent the target product as A. Consider A's entire life cycle. A's life cycle is divided into a sequence of L life phases (or *life stages* as used by some researchers) as shown in Figure 3.1. Division is based on the extent of interest and the necessary granularity.

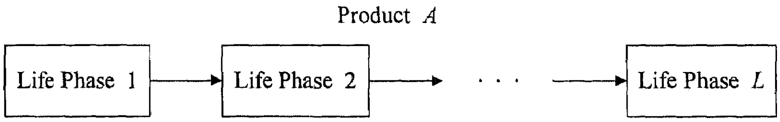


Figure 3.1 Representation of product A's life cycle.

For life phase k, its input product status is S_{k-1}, and its output product status is S_k (Figure 3.2).

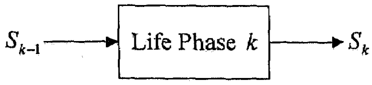


Figure 3.2 Product status representation of a life phase.

There are n_k major processes/activities in life phase k, i.e., n_k major processes/activities constitute the transformation of product status from S_{k-1} to S_k. These n_k processes/activities are represented by a process set P_k:

$$P_k = \left\{p_1^k, p_2^k, \cdots, p_{n_k}^k\right\},$$

which is constrained by process sets $P_1, P_2, \cdots,$ and P_{k-1}:

$$P_k\big|_{P_1, P_2, \cdots, P_{k-1}}.$$

Therefore, life phase k has an alternative process set representation as shown in Figure 3.3.

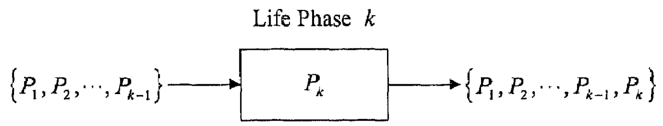


Figure 3.3 Process set representation of a life phase.

Based on the preceding representation of a product's life cycle, a general index vector $\mathbf{c}$,

$$\mathbf{c} = \begin{pmatrix} c_1 & \cdots & c_i & \cdots & c_\alpha \end{pmatrix}^T,$$

is proposed and applied to each life phase to evaluate its corresponding performance. Index vector $\mathbf{c}$ may consist of such components as cost, benefit, and environmental impact. For life phase k, we have an index vector $\mathbf{c}_k$:

$$\mathbf{c}_k = \mathbf{c}(P_k) = \begin{pmatrix} c_{1k} & \cdots & c_{ik} & \cdots & c_{\alpha k} \end{pmatrix}^T,$$

where $c_{ik} = c_i(P_k)$ is the ith index of life phase k.

Altogether, there are L index vectors (one for each life phase):

$$\mathbf{c}_1 \cdots \mathbf{c}_k \cdots \mathbf{c}_L.$$

They are gathered into an index matrix $\mathbf{C}$ to evaluate the overall performance of the product's entire life cycle:

$$\mathbf{C} = \begin{pmatrix} \mathbf{c}_1 & \cdots & \mathbf{c}_k & \cdots & \mathbf{c}_L \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1k} & \cdots & c_{1L} \\ \vdots & & \vdots & & \vdots \\ c_{\alpha 1} & \cdots & c_{\alpha k} & \cdots & c_{\alpha L} \end{pmatrix}_{\alpha \times L} .$$

Various transforms of $\mathbf{C}$ can also be defined to evaluate product A's different performances:

$$\text{transform}(\mathbf{C}) .$$

Finally, the generally concerned research issues can be expressed as:

> Given $\mathbf{C}$ or $\text{transform}(\mathbf{C})$, find process sets $P_1, P_2, \cdots,$ and P_L,
>
> such that $\mathbf{C}$ or $\text{transform}(\mathbf{C})$ is optimized.

3.3 An Example

This section provides some additional explanations about the physical meaning of the generic framework. A simple cup development example is used throughout this section to clarify the statement.

As mentioned in Chapter 1, the increasing number and variety of requirements from different life phases of a product merge into the need for an overall consideration of a product's entire life cycle and ultimately an integrated product and process development environment. For an integrated cup development environment, for example, a cup's entire life cycle should be considered. Also, with the requirements for the different life phases of a cup, we should divide a cup's life cycle into interesting life phases, with each life phase corresponding to its specific performance requirements. For a simple illustration, assume that a cup has a life cycle as shown in Figure 3.4.

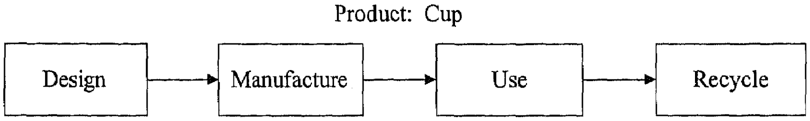


Figure 3.4 A simple example: A cup's life cycle.

In Figure 3.4, a cup's life cycle consists of four consecutive life phases, i.e., design, manufacture, use, and recycle. Each life phase has its corresponding activities and performance criteria. Moreover, there exists close relationship among these life phases. Theoretically, the activities in one life phase influence the activities in all its posterior life phases. In this example, the design phase influences the other three phases; the manufacture phase influences the use and recycle phases; and the use phase influences the recycle phase. In other words, the activities in one life phase are influenced by the activities in each of its prior life phases. As a result, if we want to *really* optimize the performance of a life phase, not only should the activities in this life phase be adjusted, the activities in each of its prior life phases should be adjusted as well. For example, in order to optimize the performance of a cup's recycle phase, we definitely need to select the best recycling method to obtain low cost, minimum environmental impact, and maximum benefit, etc. Moreover, many other more important adjustment decisions should be made in the design, manufacture, and use phases as well, which all influence the recycle phase. For example, a better design, which may mean selection of a better material with less hazardous component, can make recycling much easier and with less environmental impact.

To further quantify the preceding explanations, assume that for each life phase k in Figure 3.4, process set P_k consists of only one process p^k, i.e.,

$$P_k = \{p^k\}.$$

Each process p^k may be one of several possible selections. The possible processes for each p^k are summarized in Table 3.1.

Table 3.1 Possible processes in a cup's life cycle.

Life Phase	Process	Meaning
Design	D1	Foam cup design
	D2	Plastic cup design
	D3	Glass cup design
Manufacture	M1	Manufacture foam cup
	M2	Manufacture plastic cup
	M3	Manufacture glass cup
Use	U	Use the cup
Recycle	R1	Foam cup recycled for secondary use (e.g. packaging)
	R2	Plastic cup material recycling process A
	R3	Plastic cup material recycling process B
	R4	Glass cup material recycling process C
	R5	Glass cup material recycling process D

Assume that the general index vector $\mathbf{c}$ has two component c_1 and c_2. c_1 means cost and c_2 means environmental impact. For life phase k, we have an index vector $\mathbf{c}_k = \mathbf{c}(P_k) = \mathbf{c}(p^k)$. For a cup's entire life cycle, we have an index matrix:

$$\mathbf{C} = \begin{pmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 \end{pmatrix}.$$

Based on the former information, we can enumerate all the possible cases of a cup's life cycle as shown in the directed graph of Figure 3.5. In Figure 3.5, there are two virtual nodes (origin and terminal) and four levels of real nodes. Level k corresponds to life phase k in a cup's life cycle. A directed path from the origin to the terminal represents a possible case of a cup's life cycle. A real node on level k has marking $\{p^1,\cdots,p^k\}$ (referring to the representation method of Figure 3.3). The input arc to a real node on level k has weight $\mathbf{c}(p^k)$.

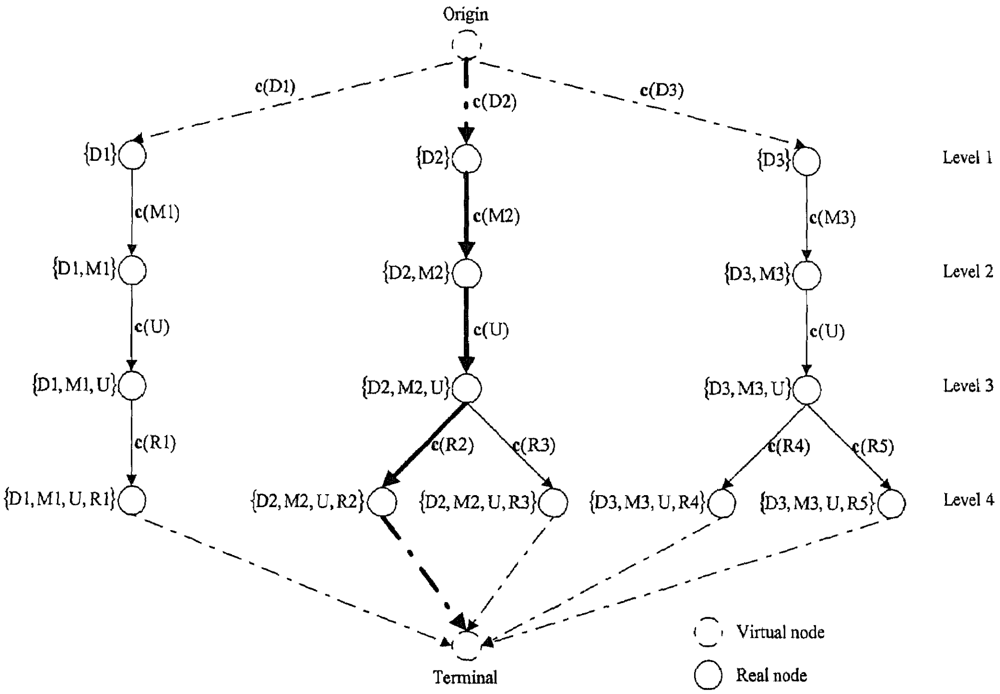


Figure 3.5 A directed graph showing all the possible cases of a cup's life cycle.

Now assume that we want to optimize the index vector $\mathbf{c}_4 = \mathbf{c}(p^4)$, which means that we want to minimize cost $c_1(p^4)$ and environmental impact $c_2(p^4)$. In order to do this, we simply check the values of $\mathbf{c}_4 = \mathbf{c}(p^4)$ at all the possible processes of p^4 and find out that recycling process R2, for example, is the optimal one. This recycling process in turn requires design D2 in the design phase, and manufacture process M2 in the manufacture phase. Thus, in order to optimize $\mathbf{c}_4 = \mathbf{c}(p^4)$, we make the following selections:

$$p^1 = \text{D2}, \quad p^2 = \text{M2}, \quad p^3 = \text{U}, \quad \text{and} \quad p^4 = \text{R2}.$$

This is shown by the bold path in Figure 3.5.

This simple product example shows the practical meaning of the proposed generic framework.

3.4 Summary

Product development is essential for the economic success of manufacturing firms. A great amount of research issues are being carried out concerning different aspects of this problem. This chapter first indicates that most of these issues have the same basic nature: performance optimization while meeting the requirements exerted on a product. Based on this nature, a generic framework for integrated product and process development is then proposed. In this framework, a product's entire life cycle is considered and divided into a sequence of life phases. Each life phase transforms the product status from one to another and is represented as a set of processes. A general index vector is used to evaluate the performance of each life phase. It may include such components as cost, benefit, and environmental impact. The index vectors (one for each life phase) are gathered into an index matrix to evaluate the overall performance of the product's entire life cycle. The index matrix and its various transforms then serve as the optimization objective for different purposes. By this generic framework, most of the concerned issues can be formally described as constrained optimization problems. Physical meaning of the generic framework is further discussed through a simple product example.

CHAPTER 4

AN IPPD METHODOLOGY: CONCEPT FORMULATION

Although some promising observations and results are obtained in Chapter 3, they are regarded as some preliminary results and the first part of the IPPD methodology. In order for them to be applicable to real product and process development, concepts and terms need be defined clearly, and application procedures are needed to illustrate how to apply these concepts and the generic framework systematically and meaningfully. These two parts of the methodology are the focus of this chapter. In Section 4.1, some basic concepts are defined, and in Section 4.2, an application procedure is provided.

4.1 Basic Concepts

Following are some basic concepts of the IPPD methodology.

(1) Expected Life Cycle Structure (ELCS)

An ELCS is the structure of the life cycle that a target product is expected to have. This structure is a general profile of the product's entire life cycle according to product developers' high level expectations. An ELCS is graphically represented using three basic elements as shown in Figure 4.1:

(i) Solid line boxes representing life phases (Figure 4.1(a));

(ii) Dotted line ellipses representing major external input/output of information or material (Figure 4.1(b));

(iii) Directed arcs connecting these two types of elements (Figure 4.1(c)).

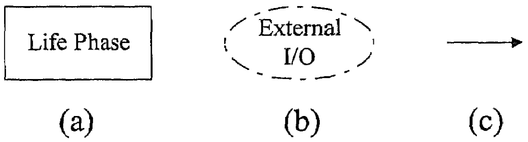


Figure 4.1 Basic elements in the graphical representations of an ELCS.

For a target product, there may be different ELCSs, different system boundaries, and different life phase granularities depending on the developers' expectation, interest, and concern.

Figure 4.2 is a typical example of a product's expected life cycle structure.

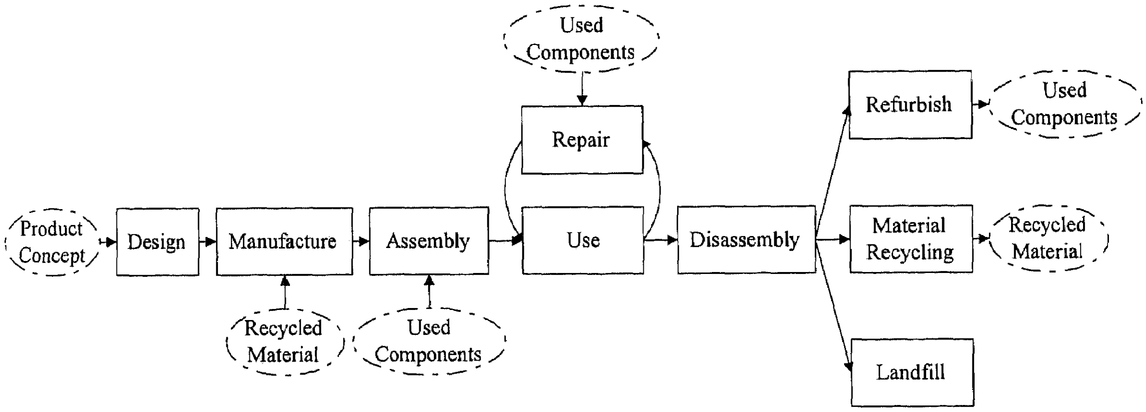


Figure 4.2 A typical ELCS example.

The product's life cycle is expected to begin with a product concept and then enter its design life phase. It then has its manufacture life phase and assembly life phase. In its manufacture life phase, it should use recycled material wherever possible; and in its assembly life phase, used components should be considered with priority. The product then enters its use life phase and repair life phase loop, where used components are considered wherever possible in the repair life phase. It then comes the disassembly life phase. After disassembly, some components enter the refurbish life phase and are output as used components; some other components enter the material recycling life phase and are output as recycled material; and the others are landfilled.

(2) ELCS Graph

Given an expected life cycle structure (ELCS), we can obtain its corresponding graph representation simply by substituting each of its boxes and ellipses with a node in the graph and keeping each arc unchanged. The obtained graph is called an ELCS graph. Figure 4.3 is the ELCS graph corresponding to the ELCS shown in Figure 4.2.

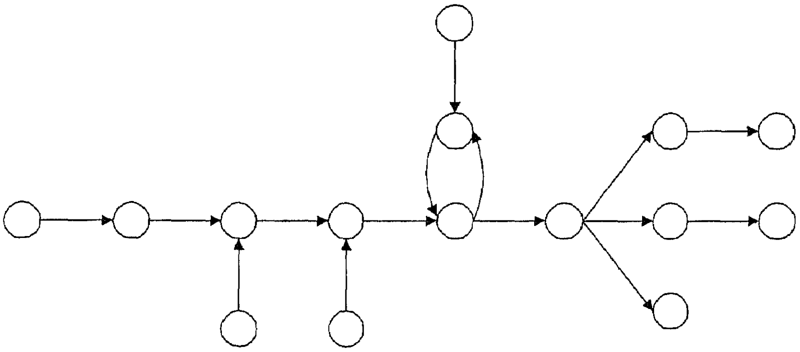


Figure 4.3 The ELCS graph corresponding to the ELCS in Figure 4.2.

An ELCS graph is a directed and connected graph. Each node has an in-degree and an out-degree. Similarly, we define the in-degree and out-degree of a sub-graph of an ELCS graph as the number of arcs going into and away from the sub-graph, respectively.

Keeping in mind that each node in the ELCS graph has a single correspondent (life phase or external I/O) in the ELCS, in later context, we will use the concepts of ELCS and ELCS graph interchangeably without making special distinctions.

(3) Expected Life Cycle (ELC) Sub-structure

A sub-graph of an ELCS graph is an ELC sub-structure iff:

(i) It is connected;

(ii) It has an in-degree and out-degree of 0 or 1; and

(iii) For any two nodes in the ELCS graph, if they are both in the sub-graph, all the directed arcs between them are in the sub-graph.

Figure 4.4 gives some examples of ELC sub-structures each within a dotted line ellipse.

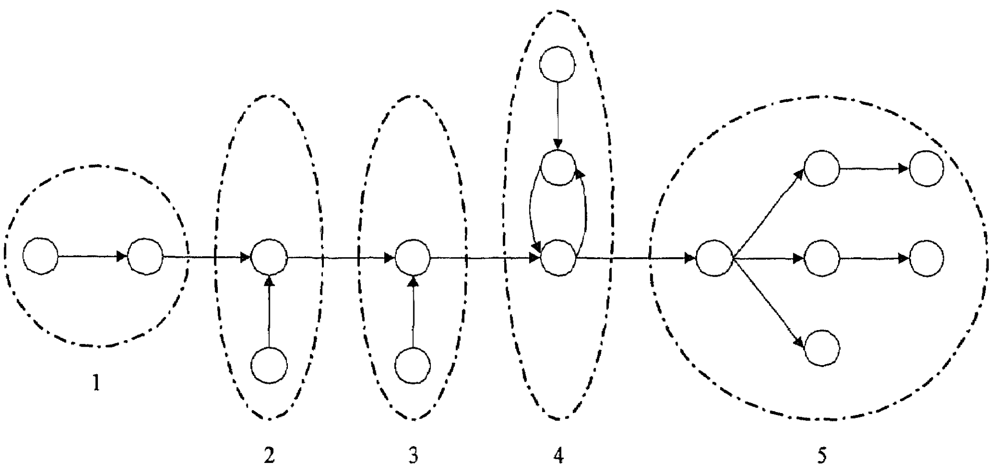


Figure 4.4 Some ELC sub-structures.

The in-degree of ELC sub-structure 1 is 0 and its out-degree is 1. The in-degrees and out-degrees of ELC sub-structures 2; 3, and 4 are all 1's. The in-degree of ELC sub-structure 5 is 1 while its out-degree is 0.

(4) Simple and Complex ELC Sub-structures

An ELC sub-structure is simple if each of its nodes has an in-degree and out-degree of 0 or 1, otherwise it is complex. Here, the in-degree and out-degree of each node refer to the node degrees calculated in the ELCS graph, not in the ELC sub-structure.

For example, consider the ELC sub-structure 2 in Figure 4.4. The node corresponding to the manufacture life phase has an in-degree of 2 and an out-degree of 1. Therefore it is a complex ELC sub-structure. Similarly, sub-structure 1 is simple, and sub-structures 3, 4, and 5 are all complex.

(5) Simple and Complex ELCSs

If an ELCS contains a complex ELC sub-structure, it is a complex ELCS, otherwise it is simple. In other words, in a simple ELCS, each node has an in-degree and out-degree of 0 or 1, while in a complex ELCS, there exist exceptions. By this definition, the ELCS shown in Figure 4.2 is a complex ELCS.

(6) Simplification of a Complex ELCS

A complex ELCS can be transformed into a simple ELCS by the following simplification procedure:

(i) Find out all the complex ELC sub-structures that are exclusive;

(ii) For each of them, if the ELC sub-structure contains a life phase, substitute it with a compound life phase; otherwise, substitute it with a compound external I/O;

(iii) Each compound life phase or compound external I/O represents a complex ELC sub-structure and includes the aggregate of activities in the ELC sub-structure.

After this, a simplified ELCS is obtained. It is a sequence of consecutive life phases probably with external I/O at its beginning or end. In order to avoid huge compounds, in the first step of the simplification procedure, we prefer to select a complex ELC sub-structure that contains fewer nodes.

As an example, consider simplification of the complex ELCS shown in Figure 4.2. It has 4 complex and exclusive ELC sub-structures as indicated in Figure 4.4, i.e., ELC sub-structures 2, 3, 4, and 5. Following the simplification procedure, we substitute each of them with a compound life phase, and thus obtain a simplified ELCS as shown in Figure 4.5.

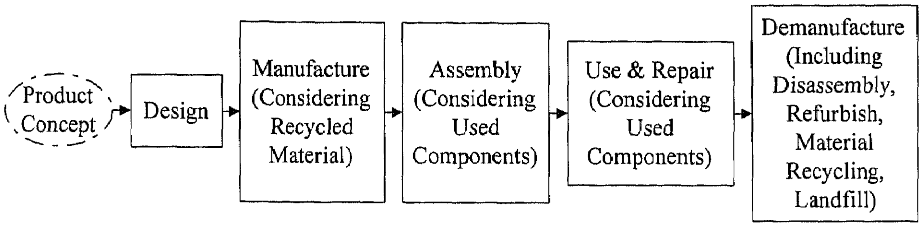


Figure 4.5 A simplified ELCS of the complex ELCS in Figure 4.2.

Excluding the external information input of product concept, the simplified ELCS is composed of 5 consecutive life phases: design, manufacture (considering recycled material), assembly (considering used components), use & repair (considering used components), and demanufacture (including disassembly, refurbish, material recycling, and landfill). The latter four are compound life phases.

The objective of obtaining a simple ELCS representation is to be able to use a tree (i.e., life locus tree) as the optimal search structure.

(7) Process

The definition and understanding of the term "process" are quite different in different areas and by different people. In the IPPD methodology, a process is defined as a basic unit of activity that is carried out during a product's life cycle. The granularity of processes depends on product developers' concern and interest, and the appearance of processes varies along with a product's entering its different life phases. For example, in a product's design life phase, a process may be a system design or a part design; in its manufacturing life phase, a process may be a specific manufacturing operation; and in its disassembly life phase, a process may be a disassembly operation, etc. In other words, a process may actually be "anything" as long as it is the activity under consideration. For product developers, the objective is therefore to select the processes in a product's entire life cycle such that the constituted life cycle is optimal or at least satisfactory.

(8) Index Vector of a Process

As mentioned before, processes are the activities constituting a product's life cycle and varying their physical meanings along with time. In order to provide a means for product developers to compare these different processes on a common basis, we propose the concept of "index vector." Index vector is a group of α indices (or features), each of which maps a process onto the real axis and represents an interesting characteristic of the process. These indices are represented as a vector called the index vector $\mathbf{c}$:

$$\mathbf{c} = \begin{pmatrix} c_1 & \cdots & c_i & \cdots & c_\alpha \end{pmatrix}^T .$$

From another viewpoint, suppose that all the possible processes under consideration for a product's life cycle are gathered into a set Ω, which is called the overall process set. Index vector $\mathbf{c}$ is then a mapping from Ω to R^α, i.e.,

$$\mathbf{c} : \Omega \mapsto R^\alpha ,$$

where R is the set of real numbers. By this representation, all the possible processes can then be compared using their corresponding index vectors.

In some cases, it is preferable that these α indices are comparable to each other as well. To do this, one can use the same unit for each index and assign weighting factors to them to represent the corresponding importance.

(9) Index Vector of a Process Set

Let P be a set of n selected processes, i.e.,

$$P = \{p_1, p_2, \cdots, p_n\},$$

where

$$p_i \in \Omega, \text{ for } i = 1, \cdots, n.$$

For each $p_i \in P$, we have its index vector $\mathbf{c}(p_i)$ evaluating its characteristics. For process set P, its index vector $\mathbf{c}(P)$ is defined as:

$$\mathbf{c}(P) = \sum_{p_i \in P} \mathbf{c}(p_i),$$

which is then used to evaluate the characteristics of the process set. Weighting factors can be assigned to the index vectors if these processes assume different significance.

(10) Classification of Requirements

Requirements for a product are classified into two types in the methodology: functional and non-functional. Functional requirements are those that a product should meet in order to still be such a "product," and non-functional requirements are those that evaluate a product as a good product or a bad one.

For example, the requirement "a cup should be microwavable" is usually considered as a functional requirement, and the requirement "a cup should not be too expensive" is usually considered as a non-functional one.

(11) Life Locus and Life Locus Tree

For a target product A, suppose that its simple or simplified ELCS, including life phases only, is as shown in Figure 4.6.

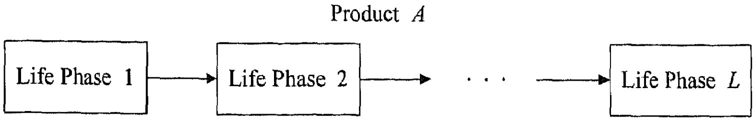


Figure 4.6 Product A's ELCS.

Keeping in mind all the functional requirements for product A, we can select the processes/activities in each life phase and obtain L process sets. These L process sets is called a life locus (LL) for product A. In other words, a life locus is a possible selection/combination of processes that constitutes a life cycle for the product following its ELCS and all the functional requirements. A life locus is represented as:

$$LL: P_1, \cdots, P_k, \cdots, P_L,$$

where P_k is the process set of life phase k $(k = 1, \cdots, L)$ and is characterized by an index vector $\mathbf{c}(P_k)$. Similarly, the index vector of a life locus is defined as:

$$\mathbf{c}(LL) = \mathbf{c}(P_1, \cdots, P_k, \cdots, P_L) = \sum_{k=1}^{L} \mathbf{c}(P_k),$$

which is used to evaluate the characteristics of the life locus. Weighting factors can be assigned to the index vectors if these process sets, i.e., life phases, assume different significance.

Moreover, for convenient representation and efficient search purposes, we set up a tree structure called "life locus tree" (T_{LL}). Figure 4.7 shows the structure of a life locus tree. The tree has $L+1$ levels. Level 0 is the root level, level L is the leaf level, and level $k\,(k = 1, \cdots, L)$ corresponds to life phase k in the product's ELCS (Figure 4.6). For each node on level $k\,(k = 1, \cdots, L)$, it represents a possible combination of process sets, i.e., $\{P_1, P_2, \cdots, P_k\}$, after life phase k (referring to Figure 3.3). For node $\{P_1, P_2, \cdots, P_k\}$ on level k, each of its children on level $k+1$ represents a possible selection of P_{k+1} constrained by $P_1, P_2, \cdots, P_k$.

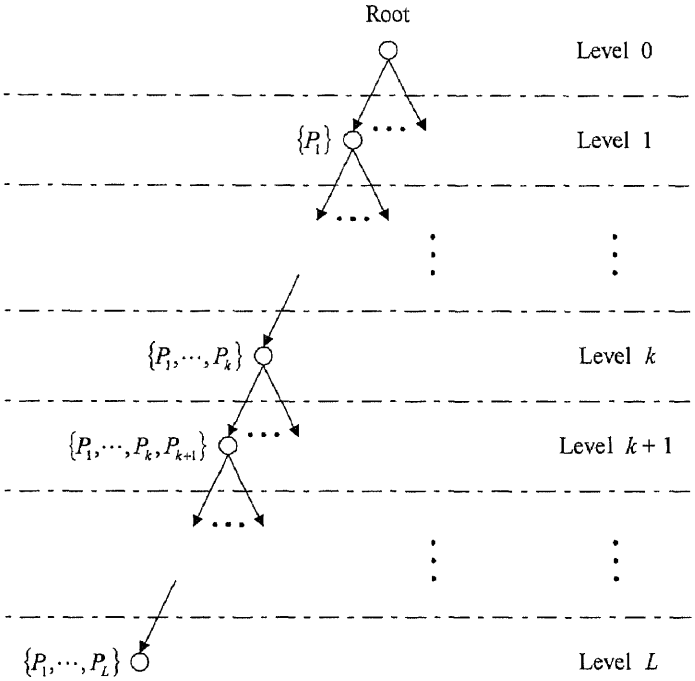


Figure 4.7 A life locus tree.

In a life locus tree, a path from the root to a leaf is a life locus. A life locus tree includes all the possible life loci that a product may have if it abides by the expected life cycle structure and all the functional requirements. Different life loci in T_{LL} can be compared by their corresponding index vectors. A variety of search algorithms (e.g., exhaustive search, heuristic search, etc.) can be used to search for an optimal life locus.

(12) Product

From the preceding explanations, it is clear that in the IPPD methodology, the concept of product has been extended to include a product's entire life cycle. In other words, a product is now identified not only by the functions it fulfils, but also by the life cycle it has. A product's life cycle is composed of a large number of processes, and the aggregate of these processes determines the overall performance of the product. The task of product development is therefore to select the processes to constitute an optimal product life cycle, i.e., an optimal product.

4.2 An Application Procedure

In Section 4.1, some basic concepts of the methodology are defined. In this section, an application procedure is provided to illustrate how to apply the methodology systematically to real product and process development.

Represent the target product as A. Identify the requirements for A and classify them into two sets: the set of functional requirements $\Re_F$ and the set of non-functional requirements $\Re_{NF}$.

An application procedure of the IPPD methodology is as follows:

Step 1: Set up an expected life cycle structure (ELCS) for product A;

Step 2: Identify whether the ELCS is simple or not. If not, simplify it;

Step 3: Based on the non-functional requirements in $\Re_{NF}$, identify a set of interesting characteristics and represent them as indices to form an index vector;

Step 4: On the basis of the knowledge about processes and their index vectors and the functional requirements in $\Re_F$, set up a life locus tree T_{LL} for product A;

Step 5: Search in T_{LL} for a life locus that meets the non-functional requirements in $\Re_{NF}$ the best.

In Step 4, we have to know all the possible processes for each life phase of the product's ELCS and the index values for each possible process. In some cases, 0's (or other specific values) can be assigned to the unknown indices.

It needs to be emphasized that non-functional requirements $\Re_{NF}$ are used when building up the index vector and searching for the optimal life locus. Functional requirements $\Re_F$ are used to identify the possible processes in the product's life cycle and set up the life locus tree. Each life locus in the life locus tree meets the functional requirements and thus is a possible candidate product. The ultimate objective of product developers is to search in these possible products (identified by their life loci) for the optimal one that meets the non-functional requirements the best.

It is easily noticed that the preceding application procedure is not like an ordinary algorithm which indicates exactly what to do. Much professional knowledge and experience may be required to accomplish the entire procedure, and some practice is needed to become familiar with it.

4.3 Summary

This chapter introduces an IPPD methodology based on the generic framework proposed in Chapter 3. The following concepts are discussed: expected life cycle structure (ELCS), ELCS graph, ELC sub-structure, simple and complex ELC sub-structures and ELCSs, simplification of a complex ELCS, process, index vector of a process and process set, classification of requirements, life locus and life locus tree, and product. An application procedure is provided to illustrate how to apply the methodology systematically to real product and process development.

Following this methodology, product developers first carry out a thorough investigation into the target product's entire life cycle and then set up an expected life cycle structure (ELCS) for the product. This ELCS is a general profile of the product's entire life cycle according to the developers' high level expectations. It may be complicated (e.g., including multi-lifecycle considerations of the product) and not suitable for representation and analysis purposes. Using a systematic approach provided in the methodology, such an ELCS can be simplified and represented as a sequence of consecutive life phases (i.e., a simple ELCS). Following this ELCS, product developers then thoroughly explore all the possible life loci for the product based on the product's functional requirements. Because of the simple nature of this ELCS (i.e., a sequence of consecutive life phases), all the possible life loci can be expressed using a tree structure, i.e., life locus tree. Finally, based on the product's non-functional requirements, a variety of search algorithms can be applied to the life locus tree to search for an optimal life locus, i.e., an optimal product.

CHAPTER 5

A CASE STUDY ON PERSONAL COMPUTER DEVELOPMENT

Chapter 4 proposes an IPPD methodology as a systematic approach to integrated product and process development. This chapter applies the methodology to personal computer development. Section 5.1 includes details of the case study, and some typical search results are provided in Section 5.2.

5.1 Personal Computer Development

This section provides a detailed case study on applying the methodology to personal computer (PC) development. For clarity, we follow the application procedure step by step.

First of all, we need to identify the requirements for a PC and classify each requirement as functional or non-functional. For the illustration purpose of this case study, we identify the functional requirements as those general definitions of a PC, and put our emphasis on the non-functional requirements. Basically we are concerned about three non-functional characteristics of a PC. First, is a PC expensive or not? Second, is a PC's performance good or bad? Third, is a PC environmentally friendly?

Step 1: Set up an expected life cycle structure (ELCS) for PC.

After analyzing the typical issues considered in a PC's lifetime, we expect without losing generality that a PC has a typical life cycle structure as shown in Figure 5.1.

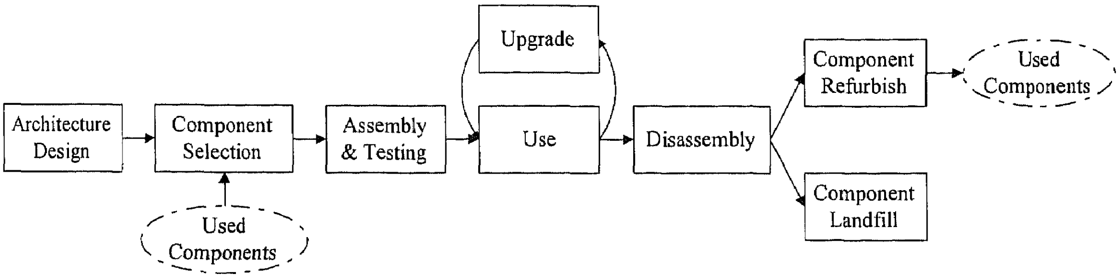


Figure 5.1 An ELCS for PC.

In this ELCS, a PC's life cycle is expected to begin with the architecture design life phase. It then enters its component selection life phase where selections of used components are considered wherever possible. After that, a PC has its assembly and testing life phase. Then it enters its use-upgrade life phase loop. After usage, a PC has its disassembly life phase. Finally some components enter the component refurbish life phase and are output as used components, and the others enter the component landfill life phase.

Step 2: Identify whether the PC's ELCS is simple or not. If not, simplify it.

The PC's ELCS shown in Figure 5.1 is a complex ELCS because there are three complex ELC sub-structures in it (Figure 5.2).

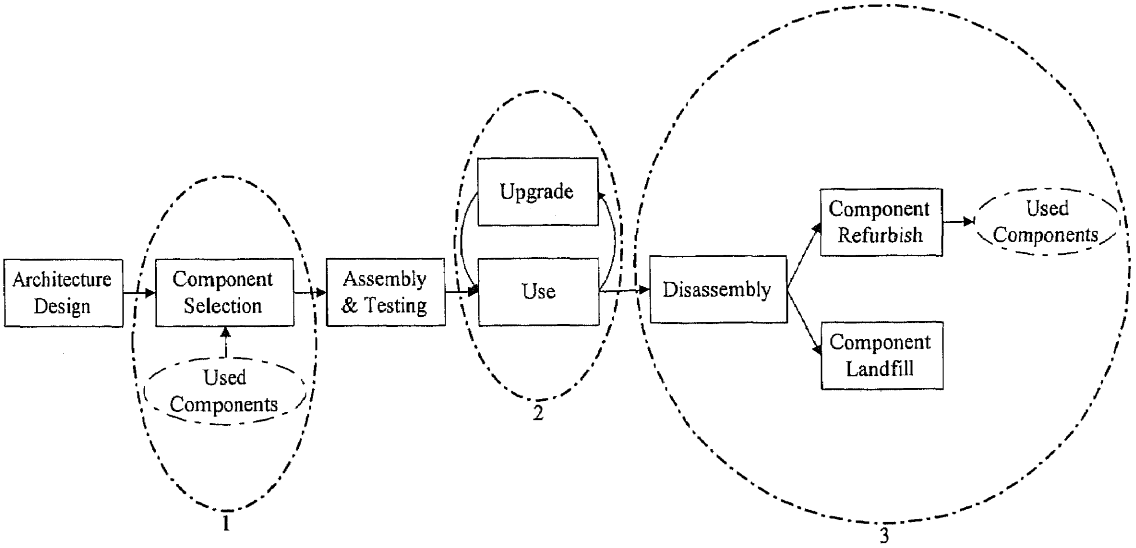


Figure 5.2 Three complex ELC sub-structures in the PC's ELCS.

For example, consider ELC sub-structure 1. Its node "component selection" has two inputs (i.e., in-degree of 2) and one output (i.e., out-degree of 1). Therefore it is a complex ELC sub-structure. Similarly, ELC sub-structures 2 and 3 are also complex.

After simplification, we obtain a simplified ELCS as shown in Figure 5.3.

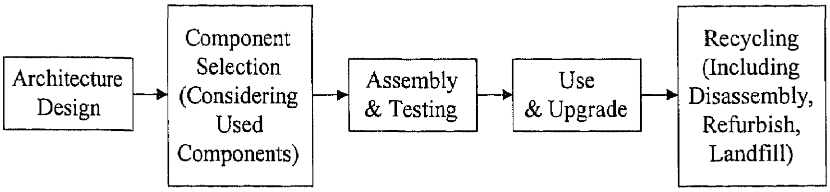


Figure 5.3 A simplified ELCS for PC.

It consists of five consecutive life phases: architecture design, component selection (considering used components), assembly & testing, use & upgrade, and recycling (including disassembly, refurbish, and landfill).

In the later context, we will use this simplified ELCS as the target ELCS.

Step 3: Based on the non-functional requirements, identify a set of interesting characteristics and represent them as indices to form an index vector.

Three interesting characteristics of a PC, i.e., non-functional requirements, are: expensive or not, good or bad performance, and environmentally friendly or not. Now we use the corresponding money equivalents of these characteristics as three indices. They are cost (c_1), benefit (c_2), and environmental impact (c_3), respectively. The index vector is then:

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} \text{Cost} \\ \text{Benefit} \\ \text{EI} \end{pmatrix},$$

where EI is the abbreviation for environmental impact. These three indices are used to evaluate the characteristics of processes, process sets (representing life phases), and ultimately a PC's entire life cycle.

Step 4: On the basis of the knowledge about processes and their index vectors and the functional requirements, set up a life locus tree T_{LL} for PC.

As shown in Figure 5.3, a PC's ELCS is composed of 5 consecutive life phases. We identify the possible processes in each life phase and get the index values for each process. Table 5.1 summarizes all the possible processes in the five life phases of a PC's ELCS.

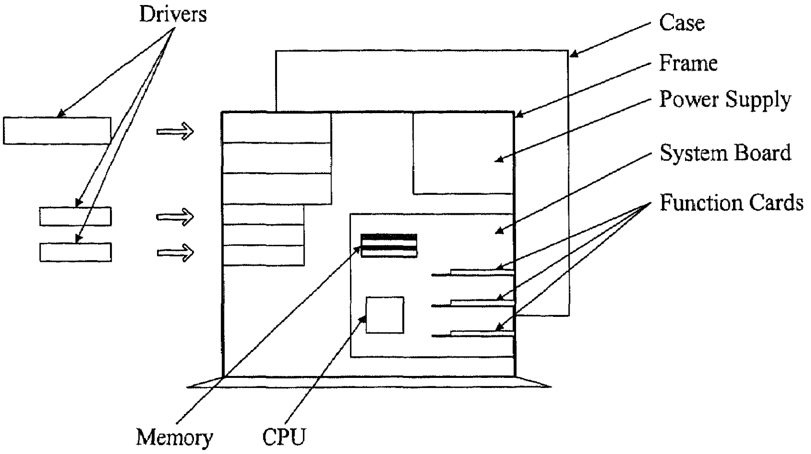
Table 5.1 Possible processes in a PC's ELCS.

Life Phase	Process Meaning	Process Set	Possible Processes
Architecture Design	A process is a PC architecture design.	$P_1 = \{p^1\}$ $n_1 = 1$	$p^1 : \begin{cases} \text{D1 : Conventional low cost PC mini - tower} \\ \text{D2 : Mini - tower PC by Compaq} \end{cases}$
Component Selection	A process is a selectable PC component.	$P_2 = \{p_1^2, \cdots, p_{n_2}^2\}$ n_2 = number of components	Each p_j^2 $(j = 1, \cdots, n_2)$ represents a selected PC component: Keyboard (KB): KB1 Monitor (MON): $\begin{cases} \text{MON1 (used 15")} \\ \text{MON2 (15")} \\ \text{MON3 (17")} \end{cases}$ Mouse (MS): MS1 CPU (CPU): $\begin{cases} \text{CPU1 (used 200MHz)} \\ \text{CPU2 (200MHz)} \\ \text{CPU3 (233MHz)} \end{cases}$ Memory (MEM): $\begin{cases} \text{MEM1 (used 16MB)} \\ \text{MEM2 (16MB)} \\ \text{MEM3 (32MB)} \end{cases}$ System board (SB): SB1 or SB2 Hard drive (HD): $\begin{cases} \text{HD1 (used 3.1G)} \\ \text{HD2 (3.1G)} \\ \text{HD3 (5.1G)} \end{cases}$ Energy management software (EMS): $\begin{cases} \text{EMS1 (without EMS)} \\ \text{EMS2 (with EMS)} \end{cases}$

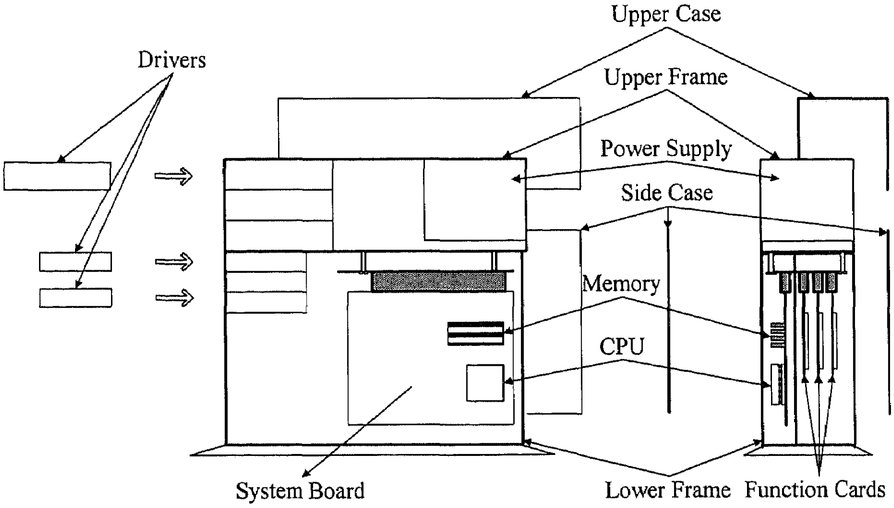
Table 5.1 (continued)

Life Phase	Process Meaning	Process Set	Possible Processes
Component Selection (continued)			Case (CASE): $\begin{cases}\text{CASE1 (used D1 case)}\\ \text{CASE2 (D1 case)}\\ \text{CASE3 (used D2 case)}\\ \text{CASE4 (D2 case)}\end{cases}$ Power supply (PS): PS1 CD-ROM (CD): CD1 Modem (MODEM): MODEM1 Sound card (SC): SC1 SVGA card (VC): VC1 Floppy disk drive (FD): FD1
Assembly & Testing	A process is an assembly sequence or a testing process.	$P_3 = \{p_1^3, p_2^3\}$ $n_3 = 2$	$p_1^3 : \begin{cases}\text{ASM1 (assembly sequence of D1)}\\ \text{ASM2 (assembly sequence of D2)}\end{cases}$ p_2^3 : TST (testing a PC)
Use & Upgrade	A process is using or upgrading a PC component.	$P_4 = \{p_1^4, \cdots, p_{n_4}^4\}$ $n_4 = 2 \times n_2$	Each PC component in P_2 corresponds to a use process and an upgrade process. For example, PC component monitor corresponds to the following two processes: Use a monitor (USEMON): $\begin{cases}\text{USEMON1}\\ \text{USEMON2}\\ \text{USEMON3}\end{cases}$ Upgrade a monitor (UPGMON): $\begin{cases}\text{UPGMON1}\\ \text{UPGMON2}\\ \text{UPGMON3}\end{cases}$
Recycling	A process is either a disassembly sequence, or the recycling process of a PC component and its possible upgraded counterpart.	$P_5 = \{p_1^5, \cdots, p_{n_5}^5\}$ $n_5 = 1 + n_2$	$p_1^5 : \begin{cases}\text{DISASM1 (disassembly sequence of D1)}\\ \text{DISASM2 (disassembly sequence of D2)}\end{cases}$ Each PC component in P_2 corresponds to a recycling process. For example, PC component monitor corresponds to the following process: Recycle a monitor and its possible upgraded counterpart (RECMON): $\begin{cases}\text{RECMON1}\\ \text{RECMON2}\\ \text{RECMON3}\end{cases}$

In the architecture design life phase, a process is a possible PC architecture design. In this case study, we consider two PC architecture designs: a conventional low cost PC mini-tower design (D1) and a mini-tower PC design by Compaq (D2) [Zhou, Caudill, and He, 1998]. These two PC architecture designs are illustrated as in Figure 5.4, where keyboard, monitor, mouse, etc. are not shown.



(a) A conventional low cost PC mini-tower design (D1)



(b) A mini-tower PC design by Compaq (D2)

Figure 5.4 Two PC architecture designs [Zhou, Caudill, and He, 1998]. (Drivers include hard drive, CD-ROM, and floppy disk drive. Function cards include sound card, SVGA card, and modem.)

In the component selection life phase, a process is a selectable PC component. For both D1 and D2, a PC consists of 15 major components: keyboard (KB), monitor (MON), mouse (MS), CPU (CPU), memory (MEM), system board (SB), hard drive (HD), energy management software (EMS), case (CASE, including case and frame), power supply

(PS), CD-ROM (CD), modem (MODEM), sound card (SC), SVGA card (VC), and floppy disk drive (FD). The capital word in the parentheses is the process name corresponding to each PC component. For example, for PC component monitor, its process name is MON and there are three different selections: MON1 is a used 15" monitor, MON2 a new 15" monitor, and MON3 a new 17" monitor. From Table 5.1, it is seen that only the following five components may be selected from used ones: monitor, CPU, memory, hard drive, and case.

In the assembly & testing life phase, a process is either an assembly sequence (the optimal sequence corresponding to a PC architecture design) or a PC testing process. For PC architecture design D1, it has assembly sequence ASM1; for D2, it has ASM2. Testing process TST is the same for both D1 and D2.

In the use & upgrade life phase, a process is either the process of using a PC component, or the process of upgrading a PC component. For each possible component x selected in life phase 2, it corresponds to a use process USEx and an upgrade process UPGx.

In the recycling life phase, a process is either a disassembly sequence (the optimal sequence corresponding to a PC architecture design) or the recycling process of a PC component and its possible upgraded counterpart. For PC architecture design D1, it has disassembly sequence DISASM1; for D2, it has DISASM2. For each component selected in life phase 2 (component selection), it may have been upgraded in life phase 4 (use & upgrade). Both the original component and its upgraded counterpart are recycled in the recycling life phase. In the recycling process, each component may be refurbished for second-hand use, or landfilled.

For simplicity, we assume that a PC component may be upgraded to a new one only. Also, we assume that a used component selected in life phase 2, when it enters the recycling life phase, will be landfilled and not refurbished. By this, we actually assume that each PC component can at most be used twice.

After identifying all the possible processes in a PC's life cycle, we need to get the index values for each process as well. In Step 3, we have identified three indices: cost, benefit, and EI (environmental impact). For processes in different PC life phases, these indices have different meanings. Table 5.2 summarizes the meaning of process indices in different life phases of a PC's life cycle.

Table 5.2 Meaning of process indices in a PC's different life phases.

Life Phase	Cost	Benefit	EI
Architecture Design	/*	/	/
Component Selection	Cost (price) of a PC component.	/	/
Assembly & Testing	Assembly cost or testing cost.	/	/
Use & Upgrade	Cost may be energy consumption of using a PC component, or the cost (price) of upgrading a PC component (including disassembly cost).	Benefit introduced through using a PC component, or increased benefit by an upgraded component.	Environmental impact during usage (mainly on the user), or increased environmental impact by an upgraded component.
Recycling	Cost may be disassembly cost, or the cost of recycling a PC component and its possible upgraded counterpart.	Benefit introduced by a used component after refurbishing.	Environmental impact during refurbishing (mainly on the natural environment).

* Cost in the architecture design life phase is assumed to be similar for different designs. Therefore it is not considered here.

The criterion of determining the index values is that the indices put in each life phase only reflect the tangible characteristics of the product in that life phase. For example, selection of a PC component in life phase 2 would incur benefit later in the use & upgrade life phase through using that component. Therefore the benefit of a PC component is only tangible in the use life phase. By our criterion, this benefit item is put in the use life phase instead of in the component selection life phase. This tangible characteristic concept can eliminate the possibility of multiple considerations of the same thing in different life phases.

To simplify our considerations without losing necessary insights, we assume that only 5 PC components are possibly upgraded in the use & upgrade life phase, i.e., monitor, CPU, memory, hard drive, and energy management software. We also assume that only 5 PC components are possibly refurbished for the second-hand use in the recycling life phase, i.e., monitor, CPU, memory, hard drive, and case.

There are two special types of processes that need more explanations: the component upgrade processes (UPGx) and the component recycling processes (RECx) where x denotes a selected PC component. We consider these two types of processes as typical, i.e., the index values of each such process are the mean index values of several general type processes. For example, a component upgrade process (UPGx) may include several possibilities: no upgrading or upgraded to different levels of components. These possibilities correspond to a probability sequence that is assumed known a priori. The indices of a component upgrade process (UPGx) are the weighted sum of the indices of each upgrading possibility, with the weights being the corresponding probability of each upgrading possibility.

Component recycling processes (RECx) are more complicated. They are typical processes of two possibilities: refurbish or landfill a component. Each possibility has its corresponding probability. In addition to recycling the original PC component selected in life phase 2, each component recycling process also includes recycling (typical process of refurbishing and landfilling) of the component's upgraded counterpart, which is also represented using probabilities. Altogether, the indices of a component recycling process are the sum of the indices of all these possibilities weighted by their corresponding probabilities.

Table 5.3 is a list of the index values of the key processes in a PC's life cycle based on the PC market of 1997 and some reasonable assumptions with reference to [Ferrer, 1997].

Table 5.3 Indices of key processes in a PC's life cycle.

Life Phase	Key Processes	Cost ($)	Benefit ($)	EI ($)
Architecture Design	D1(a conventional low cost PC mini-tower design)	0	0	0
	D2 (a mini-tower PC design by Compaq)	0	0	0
Component Selection	MON1 (used 15")	90	0	0
	MON2 (15")	270	0	0
	MON3 (17")	500	0	0
	CPU1 (used 200MHz)	130	0	0
	CPU2 (200MHz)	400	0	0
	CPU3 (233MHz)	500	0	0
	MEM1 (used 16MB)	26	0	0
	MEM2 (16MB)	80	0	0
	MEM3 (32MB)	140	0	0
	SB1 (TX QDI)	100	0	0
	SB2 (VX QDI Jumpless)	140	0	0
	HD1 (used 3.1G)	70	0	0
	HD2 (3.1G)	220	0	0
	HD3 (5.1G)	400	0	0
	CASE1 (used D1 case)	3	0	0
	CASE2 (D1 case)	10	0	0
	CASE3 (used D2 case)	5	0	0
	CASE4 (D2 case)	15	0	0
	EMS1 (without EMS)	0	0	0
	EMS2 (with EMS)	30	0	0
Assembly & Testing	ASM1 (assembly sequence of D1)	30	0	0
	ASM2 (assembly sequence of D2)	20	0	0
Use & Upgrade	USEMON1 (used 15")	80	300	140
	USEMON2 (15")	70	500	100
	USEMON3 (17")	90	800	140
	USECPU1 (used 200MHz)	12.5	700	0
	USECPU2 (200MHz)	12.5	700	0
	USECPU3 (233MHz)	12.5	900	0
	USEMEM1 (used 16MB)	12.5	120	0
	USEMEM2 (16MB)	12.5	120	0
	USEMEM3 (32MB)	12.5	180	0
	USESB1 (TX QDI)	12.5	120	0
	USESB2 (VX QDI Jumpless)	12.5	160	0

Table 5.3 (continued)

Life Phase	Key Processes	Cost ($)	Benefit ($)	EI ($)
Use & Upgrade (continued)	USEHD1 (used 3.1G)	12.5	210	0
	USEHD2 (3.1G)	12.5	240	0
	USEHD3 (5.1G)	12.5	270	0
	USEEMS1 (without EMS)	0	0	0
	USEEMS2 (with EMS)	-50	0	-40
	UPGMON1 (upg to 17" or 21")	570	200	14
	UPGMON2 (upg to 17" or 21")	570	150	40
	UPGMON3 (upg to 21")	400	35	14
	UPGCPU1 (upg to 233MHz or 266MHz)	400*	170	0
	UPGCPU2 (upg to 233MHz or 266MHz)	400*	170	0
	UPGCPU3 (upg to 266MHz)	230*	30	0
	UPGMEM1 (upg to 32MB or 64MB)	179*	138	0
	UPGMEM2 (upg to 32MB or 64MB)	179*	138	0
	UPGMEM3 (upg to 64MB)	130*	18	0
	UPGHD1 (upg to 4.0G or 5.1 G)	230*	72	0
	UPGHD2 (upg to 4.0G or 5.1 G)	230*	30	0
	UPGHD3 (upg to 6.5G)	130*	18	0
	UPGEMS1 (upg from without EMS to with EMS)	-6.7	0	-13
	UPGEMS2 (no upg)	0	0	0
Recycling	DISASM1 (disassembly sequence of D1)	50	0	0
	DISASM2 (disassembly sequence of D2)	30	0	0
	RECMON1 (used 15")	13.3	94	75
	RECMON2 (15")	23.3	139	55
	RECMON3 (17")	13.3	150	48.3
	RECCPU1 (used 200MHz)	3.3	67	4.3
	RECCPU2 (200MHz)	8.3	134	5.8
	RECCPU3 (233MHz)	6.6	122	4.7
	RECMEM1 (used 16MB)	3.3	30	3
	RECMEM2 (16MB)	8.3	43	5
	RECMEM3 (32MB)	6.7	45	4
	RECHD1 (used 3.1G)	1.7	39	5.7
	RECHD2 (3.1G)	4.2	76	6.7
	RECHD3 (5.1G)	3.3	89	5.3
	RECCASE1 (used D1 case)	0	0	2
	RECCASE2 (D1 case)	1.5	1.7	2.5
	RECCASE3 (used D2 case)	0	0	2
	RECCASE4 (D2 case)	1.5	2.5	2.5

* Partial disassembly cost: + $15 for architecture design D1; + $5 for D2.

Several guidelines are used in obtaining these values:

(i) Index values are transformed into their money equivalents (unit: US dollar $);

(ii) The price of a used component is one third the price of a new one;

(iii) We assume the following a priori probabilities:

◊ For the upgrading process of a PC component, we assume that there are at most 2 upgrading options, and each upgrading option happens with probability $1/3$. Thus the probability of no upgrading for a PC component is $1/3$ or $2/3$ depending on the number of upgrading options being 2 or 1, respectively. This is only applicable to the 5 upgradable PC components;

◊ For the recycling of a PC component, we assume that the probabilities of refurbishing and landfilling are $1/2$ each. This is only applicable to the 5 refurbishable PC components.

With the preceding knowledge about processes and their index vectors, we then set up a life locus tree for PC. From now on, we only consider the key processes for each life phase as listed in Table 5.3 and omit those processes that are the same for each life locus.

The life locus tree for PC consists of 6 levels: a root level and 5 life phase levels. Figure 5.5 shows the main structure of a PC's life locus tree. The tree has 1296 life loci that are included in two main branches corresponding to architecture design D1 and D2, respectively.

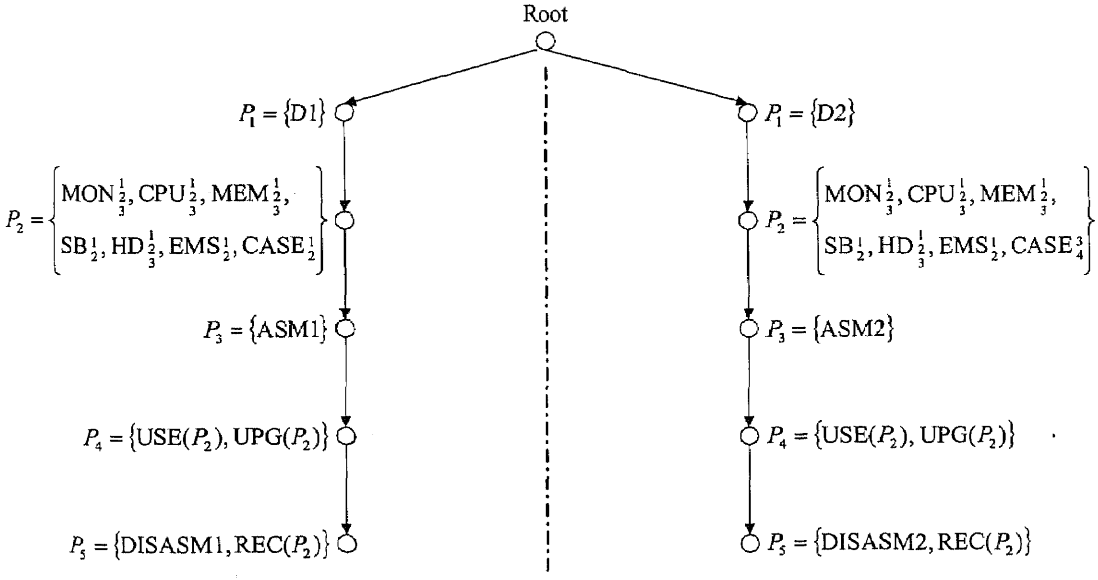


Figure 5.5 Main structure of PC's life locus tree.

Step 5: Search in T_{LL} for a life locus that meets the non-functional requirements the best.

From the analysis in the beginning of this section, we are concerned about three non-functional characteristics of a personal computer: cost, benefit, and environmental impact (EI). We hope that the cost is as low as possible, benefit as great as possible, and EI as little as possible. For the illustration purpose of this case study, we simply set the optimization objective, or search criterion C, on the life locus tree as:

$$C = w_1 \times \text{Cost} - w_2 \times \text{Benefit} + w_3 \times \text{EI},$$

where w_1, w_2, and w_3 are weights of the three indices that can be adjusted to reflect different emphasis. The item Cost in C denotes the overall cost of a life locus that is the sum of the costs of all the processes in that life locus, and the items Benefit and EI in C are similarly calculated. In other words, equal importance of processes and life phases is assumed. With this search criterion C, we then search in T_{LL} for an optimal life locus, i.e., $P_1^*, \cdots, P_5^*$. We adjust the weights w_1, w_2, and w_3 and compare the search results. Detailed search results are provided next.

5.2 Search Results and Discussions

In Section 5.1, we set the search criterion as $C = w_1 \times \text{Cost} - w_2 \times \text{Benefit} + w_3 \times \text{EI}$. By varying weights w_1, w_2, and w_3, we can get different search results (optimal life loci). Comparing these differences, valuable insights into PC designs can be obtained.

Table 5.4 lists some typical search results (i.e., optimal life loci) corresponding to different weights. When more than one life loci have the same optimal criterion value C^*, we compare their indices that are not included in C and select the ultimate optimum.

Table 5.4 Optimal life loci for a PC.

$(w_1\ w_2\ w_3)^T$	$(1\ 0\ 0)^T$	$(0\ 1\ 0)^T$	$(0\ 0\ 1)^T$	$(1\ 0\ 10)^T$
C	Cost	- Benefit	EI	Cost + 10EI
P_1^*	{D2}	{D2}	{D2}	{D2}
P_2^*	{MON1, CPU1, MEM1, SB1, HD1, CASE3, EMS2}	{MON3, CPU3, MEM2, SB2, HD3, CASE4, EMS2}	{MON2, CPU1, MEM1, SB1 (or **SB2**), HD3, CASE3, EMS2}	{MON2, CPU1, MEM1, SB1, HD1, CASE3, EMS2}
P_3^*	{ASM2}	{ASM2}	{ASM2}	{ASM2}
P_4^*	{USE(P_2), UPG(P_2)}	{USE(P_2), UPG(P_2)}	{USE(P_2), UPG(P_2)}	{USE(P_2), UPG(P_2)}
P_5^*	{DISASM2, REC(P_2)}	{DISASM2, REC(P_2)}	{DISASM2, REC(P_2)}	{DISASM2, REC(P_2)}
Cost^*	1981.6	2777.0	2393.2 (or **2433.2**)	2161.6
Benefit^*	2260.0	2877.5	2511.0 (or **2551.0**)	2455.0
EI^*	204.0	179.8	169.6	170.0
C^*	1981.6	-2877.5	169.6	3861.6

Table 5.4 (continued)

$(w_1\ w_2\ w_3)^T$	$(1\ 1\ 0)^T$	$(0\ 1\ 10)^T$	$(1\ 1\ 10)^T$	$(2\ 1\ 10)^T$
C	Cost - Benefit	- Benefit + 10EI	Cost - Benefit + 10EI	2Cost - Benefit + 10EI
P_1^*	{D2}	{D2}	{D2}	{D2}
P_2^*	{MON3, CPU1, MEM1, SB1 (or **SB2**), HD1, CASE3, EMS2}	{MON3, CPU3, MEM1, SB2, HD3, CASE3, EMS2}	{MON3, CPU1, MEM1, SB1 (or **SB2**), HD1, CASE3, EMS2}	{MON2, CPU1, MEM1, SB1, HD1, CASE3, EMS2}
P_3^*	{ASM2}	{ASM2}	{ASM2}	{ASM2}
P_4^*	{USE(P_2), UPG(P_2)}	{USE(P_2), UPG(P_2)}	{USE(P_2), UPG(P_2)}	{USE(P_2), UPG(P_2)}
P_5^*	{DISASM2, REC(P_2)}	{DISASM2, REC(P_2)}	{DISASM2, REC(P_2)}	{DISASM2, REC(P_2)}
Cost^*	2231.6 (or **2271.6**)	2706.5	2231.6 (or **2271.6**)	2161.6
Benefit^*	2651.0 (or **2691.0**)	2862.0	2651.0 (or **2691.0**)	2455.0
EI^*	177.3	177.3	177.3	170.0
C^*	-419.4	-1089.0	1353.6	3568.2

Table 5.4 (continued)

$(w_1\ w_2\ w_3)^T$	$(1\ 2\ 10)^T$	$(1\ 1\ 20)^T$
C	Cost - 2Benefit + 10EI	Cost - Benefit + 20EI
P_1^*	{D2}	{D2}
P_2^*	{MON3, CPU3, MEM1, SB2, HD1, CASE3, EMS2}	{MON2, CPU1, MEM1, SB1 (or **SB2**), HD1, CASE3, EMS2}
P_3^*	{ASM2}	{ASM2}
P_4^*	{USE(P_2), UPG(P_2)}	{USE(P_2), UPG(P_2)}
P_5^*	{DISASM2, REC(P_2)}	{DISASM2, REC(P_2)}
Cost* Benefit* EI*	2474.9 2806.0 177.7	2161.6 (or **2201.6**) 2455.0 (or **2495.0**) 170.0
C^*	-1360.1	3106.6

For example, when $w_1 = 1$, $w_2 = 0$, $w_3 = 0$, the optimal life locus that has minimum total cost is: an architecture design D2, a used 15" monitor, a used 200MHz CPU, a used 16MB memory, a cheaper system board SB1, a used 3.1G hard drive, a used D2 case, and with energy management software. When $w_1 = 0$, $w_2 = 1$, $w_3 = 0$, the optimal life locus that has maximum total benefit are: an architecture design D2, a new 17" monitor, a new 233MHz CPU, a new 16MB memory, a better system board SB2, a new 5.1G hard drive, a new D2 case, and with energy management software. When $w_1 = 0$, $w_2 = 0$, $w_3 = 1$, the optimal life loci that have minimum total environmental impact are: an architecture design D2, a new 15" monitor, a used 200MHz CPU, a used 16MB memory, a cheaper system board SB1 or a better system board SB2, a new 5.1G hard drive, a used D2 case, and with energy management software. Other results can be similarly explained referring to the process meanings in Tables 5.1 and 5.3. From these search results, we can conclude that PC architecture design D2 is better than architecture design D1.

The search algorithm used is exhaustive search. The index vector of each life locus is first calculated by summing up the index vectors of its component processes. Their criterion values are then calculated and compared to obtain the optimal one(s). The program is run on Sun SPARCstation 20. For a life locus tree with 1296 life loci as in this case study, the average search time (CPU time) is 26.83ms. When the size of the tree increases, the search time increases accordingly. Table 5.5 gives the average search time (CPU time) corresponding to increasing life locus tree size (i.e., number of life loci in the tree) as a result of more process choices in each life phase. The case with the largest number of life loci (1.3 million) takes about 28s CPU time.

Table 5.5 Average search time corresponding to increasing life locus tree size.

Size of T_{LL}	2592	5184	10368	20736	41472
Average search time (ms)	53.55	107.37	215.56	430.32	863.10
Size of T_{LL}	82944	165888	331776	663552	1327104
Average search time (ms)	1736.10	3480.00	7000.00	14010.00	28066.00

5.3 Summary

This chapter applies the IPPD methodology proposed in Chapter 4 to a real product example: personal computer development. In this application, we consider 5 consecutive life phases of a PC's ELCS (after simplification), i.e., architecture design, component selection (considering used components), assembly & testing, use & upgrade, and recycling (including disassembly, refurbish, and landfill). We consider two PC architecture designs, i.e., a conventional low cost PC mini-tower (D1) and a PC mini-tower design by Compaq (D2), and three aspects of non-functional requirements, i.e., cost, benefit, and environmental impact. We allow at most 3 different choices (or processes by our terminology) for each PC component, with each choice's characteristics known. Finally, we set up a life locus tree that has 6 levels and 1296 life loci, and successfully search for the optimal PC life loci by an exhaustive search algorithm. It is concluded that PC architecture design D2 generally outperforms D1. This case study shows that the proposed IPPD methodology can be applied to "green" product and process development meaningfully and successfully. It can be used as an important tool for green design purposes.

CHAPTER 6

INTEGRATING ECO-COMPASS CONCEPT INTO IPPD

Appropriate selection of an index vector is important to the successful application of the proposed IPPD methodology. An index vector usually consists of two parts of indices: one characterizing industrial profitability, and the other characterizing environmental impact. In the previous work, industrial profitability is conveniently evaluated using two indices: cost and benefit. However, the meaning of environmental impact is vague and the evaluation data has been reasoned without a concrete method behind it.

Motivated by the successful use of eco-compass technique for benchmark studies of existing and new products, this chapter proposes to evaluate environmental impact using eco-compass as a critical part. Section 6.1 briefly introduces the concept of eco-compass. Section 6.2 integrates eco-compass concept into the IPPD methodology. Development of a business telephone is then used as an example in Section 6.3, and Section 6.4 discusses some typical search results.

6.1 Eco-compass

Life cycle assessment (LCA) is a useful tool in analyzing the environmental impact of a product by calculating the inputs and outputs of each stage of a product's life cycle [Caudill *et al.*, 1998]. However, LCA data is usually too complex and detailed to make sense to most business decision makers. Consequently, there is a need for a means of weighting the inputs and outputs to clarify important issues and make comparisons between options.

The eco-compass technique, developed at Dow Europe, is a comparative tool to evaluate one existing product with another, or to compare a current product with new development options [Fussler and James, 1996]. The eco-compass has six dimensions, intended to encompass all significant environmental issues. Two of them are largely environmental: health and environmental potential risk, and resource conservation. Four of them are of business as well as environmental significance: energy intensity, mass intensity, revalorization, and service extension. These six dimensions are explained as follows [Fussler and James, 1996; Caudill *et al.*, 1998].

- *Mass intensity* reflects the change in the material consumption and mass burdens associated with the product over its life cycle;
- *Energy intensity* captures the change in the energy consumption associated with the product throughout its life cycle;
- *Health and environmental potential risk* detects the change in the environmental burdens associated with the product over its life cycle;
- *Revalorization* evaluates the ease with which remanufacturing, reuse, and recycling of the product can be carried out;
- *Resource conservation* detects the change in the conservation of materials and energy associated with the product over its life cycle; and
- *Service extension* measures the extent to which service can be delivered to the product throughout its life cycle.

Using eco-compass, one of the products to be compared is chosen as the base case. The base case always scores a 2 in each of six dimensions. The alternative product is then given a score relative to this base case on a scale of 0-5 in each dimension. The precise score depends on the percentage increase or decrease in performance. The diagram of eco-compass is as shown in Figure 6.1.

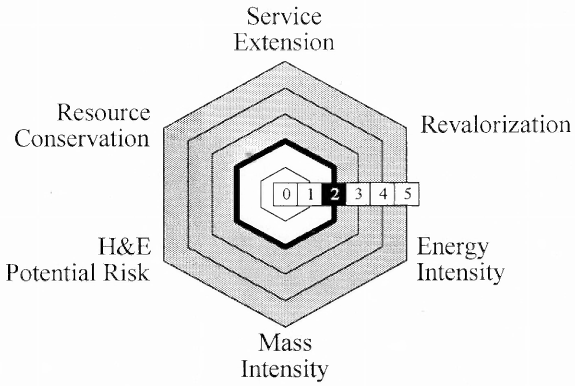


Figure 6.1 Diagram of eco-compass [Fussler and James, 1996].

6.2 Integration of Eco-compass Concept into IPPD

Eco-compass uses six dimensions to evaluate significant environmental issues associated with a product over its life cycle. Therefore we propose to use these six dimensions as six indices in the index vector of our proposed IPPD methodology to provide a more detailed and precise evaluation of a product's environmental impact. Plus two other generally used indices evaluating industrial profitability, i.e., cost and benefit, the index vector **c** then consists of eight indices:

$$\mathbf{c} = \begin{pmatrix} \text{Mass Intensity} \\ \text{Energy Intensity} \\ \text{H \& E Potential Risk} \\ \text{Revalorization} \\ \text{Resource Conservation} \\ \text{Service Extension} \\ \text{Cost} \\ \text{Benefit} \end{pmatrix}.$$

These eight indices are then used to evaluate the performance of processes, life phases, and a product's different life loci.

In order to apply this eight-index vector systematically to real product and process development, it is important to provide rules on how to evaluate each index. Figure 6.2 shows a generic model of a life phase in a product's expected life cycle structure (ELCS).

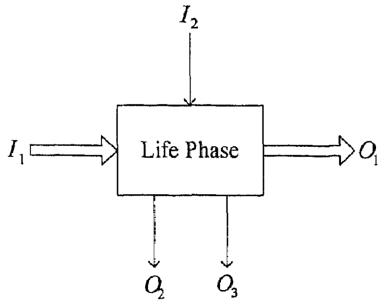


Figure 6.2 Generic model of a life phase.

We characterize two inputs and three outputs for each life phase.

- I_1: Input from other life phases. In a simple ELCS, i.e., a structure consisting of a sequence of consecutive life phases, this means input from the previous life phase.
- I_2: Input from sources other than life phases. It is divided into three parts:

$$I_2 = I_{2M} + I_{2E} + I_{2O}.$$

I_{2M} : Input of materials,

I_{2E} : Input of energy, and

I_{2O} : Input of other forms.

- O_1: Output to other life phases. In a simple ELCS, this means output to the next life phase.
- O_2: Positive (good) output to places other than life phases. It is divided into three parts:

$$O_2 = O_{2M} + O_{2E} + O_{2O}.$$

O_{2M} : Output of materials,

O_{2E} : Output of energy, and

O_{2O} : Output of other forms.

- O_3 : Negative (bad) output to places other than life phases.

Based on this generic model of a life phase, we can then evaluate the eight indices in $\mathbf{c}$ as below.

(1) Mass Intensity = I_{2M} ,

(2) Energy Intensity = I_{2E},

(3) H & E Potential Risk = O_3,

(4) Revalorization = Ease with which remanufacturing and disassembly can be carried out,

(5) Resource Conservation = $(I_{2M} - O_{2M}) + (I_{2E} - O_{2E})$,

(6) Service Extension = Ease with which service can be delivered to a product,

(7) Cost = I_{2O} , and

(8) Benefit = O_{2O}.

For general considerations, the index of revalorization is only applicable to remanufacturing, recovery, or recycling related processes and life phases, and the index of service extension is only applicable to processes and life phases related to product use.

From the evaluation equations above, it is noticed that index vector $\mathbf{c}$ is only concerned with the interactions between the product's life phases and their surrounding environment. Flows inside each life phase or among the life phases are not considered in $\mathbf{c}$ and are determined by process selections in the life phases.

Using this eight-index vector $\mathbf{c}$, we can then follow the methods and procedures in the IPPD methodology to search for an optimal life locus for a target product.

It needs to be mentioned that the generic model of Figure 6.2 can also be used to define index vectors of other forms. For example, if

$$\mathbf{c} = \begin{pmatrix} \text{Cost} \\ \text{Benefit} \\ \text{EI} \end{pmatrix},$$

where EI represents environmental impact, these three indices can be evaluated as: Cost = I_2, Benefit = O_2, and EI = O_3.

6.3 A Case Study on Business Telephone Development

This section applies the extended IPPD methodology to business telephone development. Eco-compass LCA data is provided by an LCA research group at MERC [Caudill *et al.*, 1998; Al-Okush, 1998]. This example shows how the proposed approach uses LCA data and works effectively in real applications.

(a) Expected Life Cycle Structure

First of all, we need to set up an expected life cycle structure (ELCS) for business telephone. For the illustration purpose of this case study, we apply a coarse granularity to life phases and expect a business telephone to have a typical life cycle structure as shown in Figure 6.3.

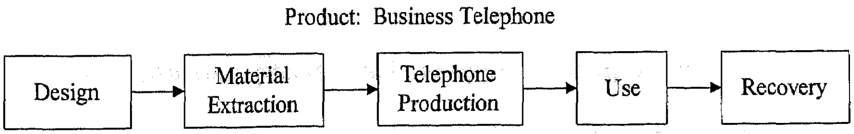


Figure 6.3 An ELCS for business telephone.

It has five consecutive life phases: (1) Design; (2) Material Extraction; (3) Telephone Production; (4) Use; and (5) Recovery.

(b) Processes

We then have to identify the possible processes in each life phase of the telephone's ELCS. A coarse granularity is applied to processes in each life phase.

In Design phase, there are four different telephone designs: D1 (1965), D2 (1978), D3 (1989), and D4 (1997) [Caudill *et al.*, 1998; Al-Okush, 1998]. We assume that they are designed to meet the same functional requirements. These four designs are as shown in Figure 6.4.

Figure 6.4 Four designs of business telephones [Caudill *et al.*, 1998; Al-Okush, 1998]. (Upper left: D1, upper right: D2, lower left: D3, lower right: D4.)

In Material Extraction phase, each telephone design D*i* may select a material extraction process from two possibilities, ME*i*1 and ME*i*2. They use different processing technologies and have different performances. Each process generates all the materials needed in a telephone, i.e., plastics, steel, aluminum, copper, etc.

In Telephone Production phase, each design Di has two alternative production processes, TP$i1$ and TP$i2$. They correspond to the use of different production equipment and therefore have different characteristics.

In Use phase, Di may be used in two different patterns, U$i1$ and U$i2$. They correspond to different utilization frequencies of the telephone and bear different characteristics.

In Recovery phase, two recovery processes are considered for each telephone design Di. R$i1$ means shredding/separation of a telephone for material and energy recovery, and R$i2$ means landfilling.

(c) Indices

A process in a telephone's life cycle is characterized by eight indices, i.e., mass intensity, energy intensity, H & E potential risk, revalorization, resource conservation, service extension, cost, and benefit. Revalorization is only applicable to recovery processes Rij $(i=1,\cdots,4; j=1,2)$, and service extension is only applicable to use processes Uij $(i=1,\cdots,4; j=1,2)$.

Applying the concept of tangible characteristics, we calculate the index values for each process. The data for processes ME$i1$, TP$i1$, U$i1$, and R$i1$ $(i=1,\cdots,4)$ is provided by MERC's LCA research group [Caudill *et al.*, 1998; Al-Okush, 1998], and the indices for ME$i2$, TP$i2$, U$i2$, and R$i2$ $(i=1,\cdots,4)$ are calculated based on the following assumptions about each pair of processes:

- ME$i1$ and ME$i2$

The yield rate for material extraction process ME$i1$ is 95% [Al-Okush, 1998]. The yield rate for ME$i2$ is 98%, the energy consumption for ME$i2$ is 10% greater than ME$i1$, its H & E potential risk is 2% less, and its cost is 5% greater than ME$i1$.

- TP$i1$ and TP$i2$

Telephone production process TP$i2$ consumes 10% more energy than TP$i1$, its H & E potential risk is 10% greater than TP$i1$, and its cost is 20% less than TP$i1$.

- U$i1$ and U$i2$

The utilization factor of usage process U$i1$ is 3% [Al-Okush, 1998]. U$i2$ has a utilization factor of 12%. Consequently, U$i2$ consumes more energy than U$i1$, and incurs more H & E potential risk. The benefit of U$i2$ is 3 times that of U$i1$.

- R$i1$ and R$i2$

R$i1$ means shredding/separation of the telephone. Metals in the telephone are recovered for reuse, and plastics are recovered for its embodied energy. R$i2$ simply landfills the telephone. Therefore the energy consumption for R$i2$ is 0, its revalorization index is 0 because remanufacturing, reuse, or recycling does not happen, and its benefit is also 0. The only cost for R$i2$ comes from the cost of the space needed in the landfill and transportation. It is calculated based on the weight of the telephone. The H & E potential risk for R$i2$ is much greater than R$i1$, and it is also calculated based on the weight.

With these reasonable assumptions and the real data provided in [Al-Okush, 1998], we list the index values for all the possible processes in a telephone's life cycle as in Table 6.1. The following abbreviations for indices are used: MI (Mass Intensity), EI (Energy Intensity), H&E (H & E Potential Risk), RV (Revalorization), RC (Resource

Conservation), and SE (Service Extension). Indices RV, SE, and Benefit are the higher the better, and all the other indices are the lower the better. The unit of indices is US dollar ($).

Table 6.1 Indices for processes in a telephone's life cycle.

Process	MI	EI	H&E	RV	RC	SE	Cost	Benefit
D1	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0
ME11	13.20	2.25	6.28	0	15.45	0	20.00	0
ME12	12.90	2.48	6.15	0	15.38	0	21.00	0
ME21	13.40	2.31	3.57	0	15.71	0	15.00	0
ME22	13.00	2.54	3.50	0	15.54	0	15.75	0
ME31	11.80	0.86	1.25	0	12.66	0	10.00	0
ME32	11.40	0.95	1.23	0	12.35	0	10.50	0
ME41	12.50	1.39	1.95	0	13.89	0	5.00	0
ME42	12.10	1.53	1.91	0	13.63	0	5.25	0
TP11	0	16.20	36.23	0	16.20	0	10.00	0
TP12	0	17.82	39.85	0	17.82	0	8.00	0
TP21	0	8.33	18.63	0	8.33	0	8.00	0
TP22	0	9.16	20.49	0	9.16	0	6.40	0
TP31	0	3.94	8.85	0	3.94	0	6.00	0
TP32	0	4.33	9.74	0	4.33	0	4.80	0
TP41	0	3.89	8.70	0	3.89	0	4.00	0
TP42	0	4.28	9.57	0	4.28	0	3.20	0
U11	0	6.50	14.56	0	6.50	35.00	0	2100.00
U12	0	26.00	58.24	0	26.00	35.00	0	6300.00
U21	0	6.50	14.56	0	6.50	40.00	0	2100.00
U22	0	26.00	58.24	0	26.00	40.00	0	6300.00
U31	0	45.31	101.43	0	45.31	110.00	0	525.00
U32	0	51.20	114.62	0	51.20	110.00	0	1575.00
U41	0	45.31	101.43	0	45.31	130.00	0	525.00
U42	0	51.20	114.62	0	51.20	130.00	0	1575.00
R11	0	1.39	3.12	5.50	-7.55	0	0	0
R12	0	0	116.25	0	0	0	2.33	0
R21	0	0.97	2.20	17.00	-9.94	0	0	0
R22	0	0	82.00	0	0	0	1.64	0
R31	0	0.50	1.11	26.25	-4.38	0	0	0
R32	0	0	41.20	0	0	0	0.82	0
R41	0	0.64	1.46	36.50	-7.19	0	0	0
R42	0	0	54.25	0	0	0	1.09	0

(d) Life Locus Tree

Based on the data and assumptions made, we can set up a life locus tree for the telephone as shown in Figure 6.5. For a clear illustration, on each level of the tree, details are only shown for a selected branch.

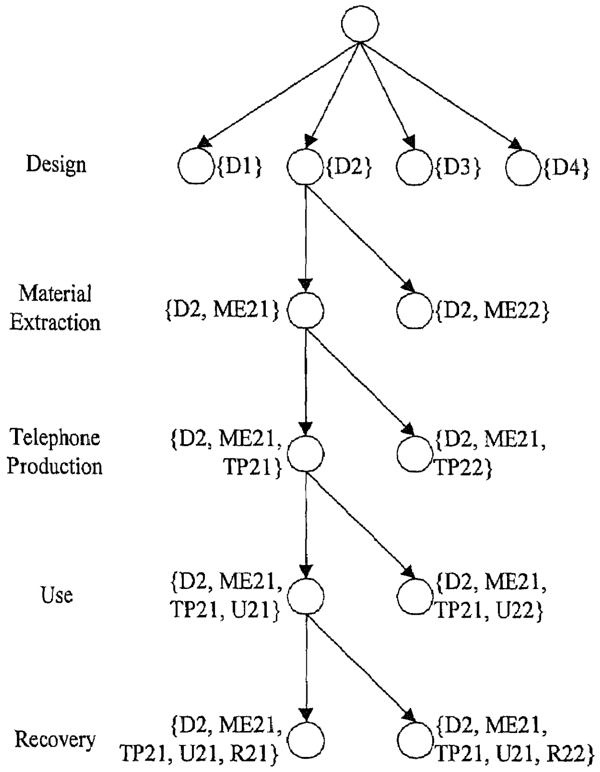


Figure 6.5 A life locus tree for telephone.

This tree has 64 life loci, each can be represented as:

$$LL_k = \{\mathrm{D}i, \mathrm{ME}ij_1, \mathrm{TP}ij_2, \mathrm{U}ij_3, \mathrm{R}ij_4\},$$

where

$$k = 1, \cdots, 64,$$
$$i = 1, \cdots, 4,$$
$$j_1, j_2, j_3, j_4 = 1, 2,$$

and

$$k = 16(i-1) + 8(j_1 - 1) + 4(j_2 - 1) + 2(j_3 - 1) + j_4 .$$

The index vector for life locus LL_k is calculated as:

$$\mathbf{c}(LL_k) = \mathbf{c}(\mathrm{D}i) + \mathbf{c}(\mathrm{ME}ij_1) + \mathbf{c}(\mathrm{TP}ij_2) + \mathbf{c}(\mathrm{U}ij_3) + \mathbf{c}(\mathrm{R}ij_4) .$$

(e) Search

We can search in the life locus tree for an optimal life locus. In this case study, we set the optimization criterion as:

$$\begin{aligned} C = {} & w_1\mathrm{MI} + w_2\mathrm{EI} + w_3\mathrm{H\,\&\,E} - w_4\mathrm{RV} \\ & + w_5\mathrm{RC} - w_6\mathrm{SE} + w_7\mathrm{Cost} - w_8\mathrm{Benefit} \end{aligned} .$$

The weighting vector $\mathbf{w} = \begin{pmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 & w_8 \end{pmatrix}^T$, and $w_i \geq 0$ $(i = 1, \cdots, 8)$. By selecting different weighting factors, we can obtain different optimal search results.

6.4 Search Results and Discussions

The optimization criterion for a telephone's life locus is defined in Section 6.3. Different weighting vectors lead to different optimal life loci, i.e., optimal telephone design and its associated production, usage, and recovery processes. An exhaustive search algorithm is used. When more than one life loci correspond to the same optimal criterion value, we simply select the one that comes first in the search procedure.

Table 6.2 provides some typical search results. They are based on different optimization criteria: some based on an individual index, and others based on combinations of all eight indices.

Table 6.2 Optimal life loci for a telephone.

w	$(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$	$(0\ 1\ 0\ 0\ 0\ 0\ 0\ 0)^T$	$(0\ 0\ 1\ 0\ 0\ 0\ 0\ 0)^T$	$(0\ 0\ 0\ 1\ 0\ 0\ 0\ 0)^T$
C	MI	EI	H&E	- RV
Optimal Life Locus	D3 ME32 TP31 U31 R31	D2 ME21 TP21 U21 R22	D2 ME22 TP21 U21 R21	D4 ME41 TP41 U41 R41
C^*	11.4	17.14	38.89	-36.5
w	$(0\ 0\ 0\ 0\ 1\ 0\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ 0\ 1\ 0\ 0)^T$	$(0\ 0\ 0\ 0\ 0\ 0\ 1\ 0)^T$	$(0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T$
C	RC	- SE	Cost	- Benefit
Optimal Life Locus	D2 ME22 TP21 U21 R21	D4 ME41 TP41 U41 R41	D4 ME41 TP42 U41 R41	D1 ME11 TP11 U12 R11
C^*	20.43	-130	8.2	-6300
w	$(1\ 1\ 1\ 1\ 1\ 1\ 0\ 0)^T$	$(1\ 1\ 1\ 2\ 1\ 2\ 0\ 0)^T$	$(0\ 0\ 1\ 2\ 1\ 2\ 1\ 0)^T$	$(0\ 0\ 1\ 0\ 1\ 0\ 1\ 1)^T$
C	MI + EI + H&E - RV + RC – SE	MI + EI + H&E - 2RV + RC - 2SE	H&E - 2RV + RC - 2SE + Cost	H&E + RC + Cost - Benefit
Optimal Life Locus	D2 ME22 TP21 U21 R21	D4 ME42 TP41 U41 R41	D4 ME42 TP41 U41 R41	D2 ME21 TP21 U22 R21
C^*	33.66	-100.39	-154.61	-6154.26

For example, in order to minimize mass intensity, we should select: D3, ME32, TP31, U31, and R31. The optimal life locus that has the best revalorization and service extension performance is: D4, ME41, TP41, U41, and R41. The optimal life locus with minimum H & E potential risk is: D2, ME22, TP21, U21, and R21. This life locus is also the optimal one when all six eco-compass indices are considered with equal importance, i.e., $\mathbf{w} = (1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0)^T$. However, if revalorization and service extension assume more significance, i.e., $\mathbf{w} = (1\ \ 1\ \ 1\ \ 2\ \ 1\ \ 2\ \ 0\ \ 0)^T$, the optimal life locus is: D4, ME42, TP41, U41, and R41. When $\mathbf{w} = (0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1)^T$, the optimization criterion $C = I_2 - O_2 + O_3$ (referring to Figure 6.2), and the corresponding optimal life locus is: D2, ME21, TP21, U22, and R21.

These results complement the previous work of [Caudill *et al.*, 1998; Al-Okush, 1998], where no decisive conclusions were provided due to the inherent difficulty of using six dimensions.

It needs to be mentioned that the validity of these search results depends on the validity of the assumptions made and LCA data provided, and the weighting factors used.

6.5 Summary

This chapter integrates eco-compass concept into the IPPD methodology to evaluate environmental impact. Eco-compass consists of six indices: mass intensity, energy intensity, health and environmental potential risk, revalorization, resource conservation, and service extension. Plus cost and benefit, an eight-index vector is set up to evaluate the performance of processes, life phases, and a product's different life loci.

As a case study, we consider the development of a business telephone. By applying the proposed approach and eco-compass LCA data provided by MERC's LCA research group, we can select the optimal telephone design and its associated production, usage, and recovery processes.

The contributions of this chapter are two-fold. One is to integrate the concept of eco-compass into the IPPD methodology, thus enrich the methodology with respect to concrete evaluation of environmental impact of a product and all related processes in its life cycle. Second, this chapter presents a detailed product example and data that can serve as a benchmark study example for different IPPD approaches.

CHAPTER 7

A SOLUTION METHODOLOGY FOR IPPD BASED ON LOGICAL REPRESENTATION OF PROCESS RELATIONS

There are two open problems with the proposed IPPD methodology. One is that simplification of a complex ELCS may not be performed efficiently for every case. Since each complex ELC sub-structure is merged into a compound life phase during simplification, the compound life phases may become too huge to be analyzed efficiently. Thus, it is highly desirable to solve the optimization problem based on the original ELCS.

Another problem is that a large amount of developer interaction is required in setting up a life locus tree. Relations among processes in a product's life cycle are an important part of product developers' knowledge and expertise. In the IPPD methodology, the developers have to consider fully the process relations and provide manually all the feasible combinations of processes for further optimization. This procedure is not trivial and its accuracy may not be guaranteed when the problem size increases.

To cope with these two problems, this chapter extends the IPPD methodology such that relations among processes in a product's life cycle are formally represented by logic. By this approach, product developers need only consider all the logical relations among processes. A logical deduction algorithm is then used to generate automatically all the feasible life loci for a target product. This approach can be applied to both simple and complex ELCSs.

This chapter is organized as follows. Section 7.1 formulates the basis of logical representation of process relations. Section 7.2 introduces a logical deduction algorithm to perform automatic life locus setup. Algorithm implementation is discussed in Section 7.3. Section 7.4 provides two application examples, one with a simple ELCS and the other with a complex ELCS.

7.1 Logical Representation of Process Relations

In this section, we first identify the basic pattern of relations among life phases in an expected life cycle structure (ELCS). Then we discuss the possible logical relations among processes. Finally we define a knowledge base of logical relations.

7.1.1 Basic Pattern of Relations among Life Phases

As defined in Chapter 4, an ELCS is the structure of the life cycle that a target product is expected to have. Basic elements of an ELCS are life phases represented as solid line boxes, external I/O of information/material represented as dotted line ellipses, and directed arcs connecting them. Figure 7.1 shows an ELCS example.

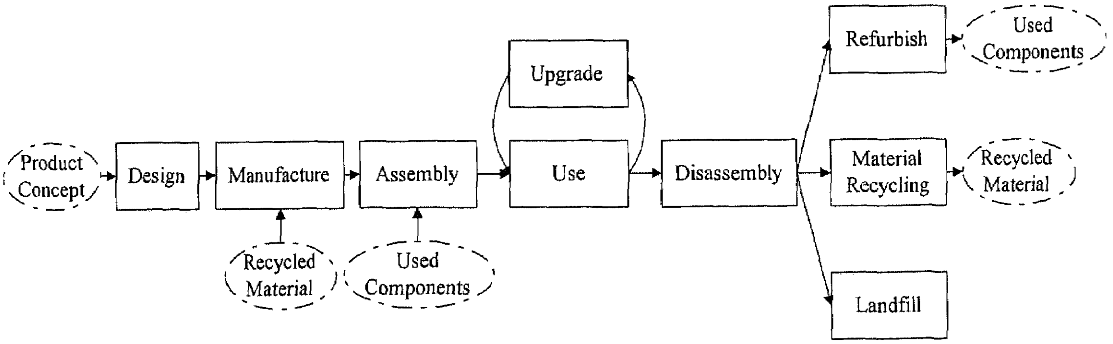


Figure 7.1 An ELCS example.

Given two life phases i_1 and i_2 in an ELCS. If there is a directed path from i_1 to i_2, the processes in i_1 may influence the processes in i_2. For the ELCS in Figure 7.1, the Design life phase has directed paths to all the other life phases. Therefore, its processes influence the processes in every other life phase. The Use life phase has directed paths to the Upgrade, Disassembly, Refurbish, Material Recycling, and Landfill life phases. Thus, its processes influence the processes in the mentioned five life phases.

7.1.2 Logical Relations among Processes

Consider an ELCS with L life phases (not necessarily consecutive). Suppose that all the possible processes under consideration for each life phase are gathered into an overall process set Ω. A life locus LL for a target product is a mapping from Ω to the set of $\{\text{True}, \text{False}\}$, i.e.,

$$LL : \Omega \mapsto \{\text{True}, \text{False}\},$$

where

$$LL(p) = \begin{cases} \text{True} & p \text{ is selected in } LL \\ \text{False} & p \text{ is not selected in } LL \end{cases} \quad \forall p \in \Omega .$$

We can establish five types of logical relations among processes in Ω:

(1) *Negation* - means that some event should not happen. This relation is represented using the logical operator $\neg$ (not);

(2) *Conjunction* - means that two events should both happen. This relation is represented using the logical operator $\wedge$ (and);

(3) *Disjunction* - means that at least one of two events should happen. This relation is represented using the logical operator $\vee$ (or);

(4) *Conditional* - means that if one event happens, the other must also happen. This relation is represented using the logical operator $\rightarrow$ (implies);

(5) *Single-selection* - means that one and only one event from a group of events should happen. No ordinary logical operators exactly match this relation, therefore we define a *single-selection logical operator* $\uparrow$ as follows:

$$A_1 \uparrow A_2 \uparrow \cdots \uparrow A_n = \bigvee_{j=1}^{n}\left(\left(\bigwedge_{\substack{i=1\\ i\neq j}}^{n} \neg A_i\right)\wedge A_j\right).$$

The logical relations among processes in Ω are then conveniently represented using these five logical operators: $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\uparrow$.

7.1.3 Knowledge Base of Logical Relations

After identifying all the logical relations among processes in a product's life cycle, a knowledge base KB can be set up. Each knowledge item in KB is a logical expression describing a certain constraint on process selections for the product. Knowledge items in KB are organized in a list as shown in Figure 7.2.

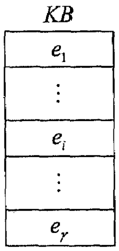


Figure 7.2 Knowledge base KB.

Suppose that KB has γ knowledge items: $e_1, e_2, \cdots$, and e_γ. Each knowledge item e_i $(i = 1, \cdots, \gamma)$ has to be satisfied in order to obtain a feasible life locus for KB. Therefore KB is represented logically as:

$$KB = e_1 \wedge e_2 \wedge \cdots \wedge e_\gamma .$$

In this work, we assume that a given knowledge base is complete, i.e., it contains all the necessary knowledge items.

7.2 Automatic Life Locus Setup

Given a knowledge base KB, the objective is to generate automatically all the feasible life loci for KB. Each feasible life locus satisfies all the logical relations in KB, i.e., bearing a True value for $KB = e_1 \wedge e_2 \wedge \cdots \wedge e_\gamma$. In this section, we propose a logical deduction algorithm to perform automatic life locus setup. Its correctness and completeness are proved and complexity analysis is conducted.

7.2.1 Preliminary

Definition 1: For $\forall p \in \Omega$, p and $\neg p$ are *primitive* logical expressions.

Definition 2: A knowledge base is *primitive* if its knowledge items are all primitive logical expressions.

Definition 3: *Standard* logical expressions are defined as below:

(1) For $\forall p \in \Omega$, p and $\neg p$ are standard;

(2) If e_1 and e_2 are standard, $e_1 \wedge e_2$ and $e_1 \vee e_2$ are standard.

Definition 4: A knowledge base is *standard* if its knowledge items are all standard logical expressions.

For a given knowledge base, its logical expressions may contain five logical operators: $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\uparrow$. We can standardize each logical expression by applying the following equivalent transforms:

- $A \rightarrow B = \neg A \vee B$;

- $A_1 \uparrow A_2 \uparrow \cdots \uparrow A_n = \bigvee_{j=1}^{n}\left(\left(\bigwedge_{\substack{i=1\\ i\neq j}}^{n} \neg A_i\right) \wedge A_j\right)$; and
- De Morgan's laws: $\neg(A \vee B) = \neg A \wedge \neg B$ and $\neg(A \wedge B) = \neg A \vee \neg B$.

The obtained standard knowledge base is equivalent to the original one.

7.2.2 Logical Deduction Algorithm

The logical deduction algorithm applies a sequence of equivalent transforms to a standard knowledge base and finds all the feasible life loci for it. The algorithm has two major steps: Process Step and Post-process Step.

Process Step:

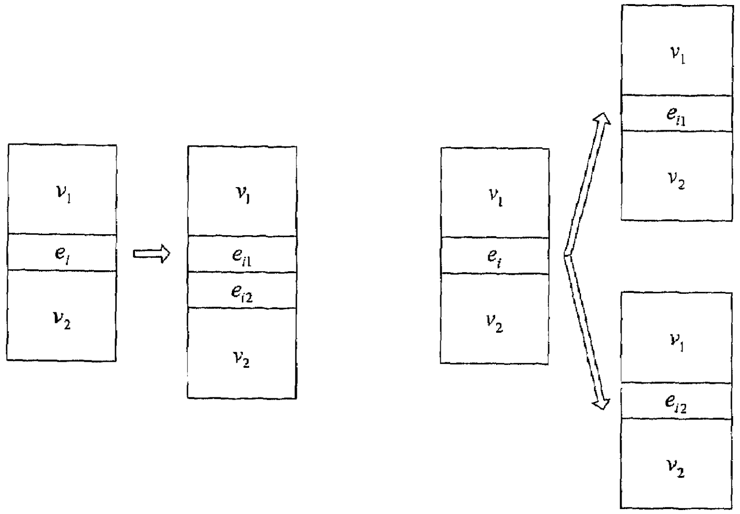
Given a standard knowledge base $KB = e_1 \wedge e_2 \wedge \cdots \wedge e_\gamma$, define U and V as two sets of knowledge bases. Initially, let $U = \Phi$ and $V = \{KB\}$.

While $V \neq \Phi$, do the following:

(1) Obtain a knowledge base $v \in V$;

(2) If v is primitive, let $U = U \cup \{v\}$ and $V = V - \{v\}$; Otherwise, suppose that knowledge item e_i in $v = v_1 \wedge e_i \wedge v_2$ is not primitive, where v_1 is either empty or contains only primitive logical expressions, then:

- If $e_i = e_{i1} \wedge e_{i2}$, let $V = V - \{v\} \cup \{v_1 \wedge e_{i1} \wedge e_{i2} \wedge v_2\}$ (Figure 7.3(a)).
- If $e_i = e_{i1} \vee e_{i2}$, let $V = V - \{v\} \cup \{v_1 \wedge e_{i1} \wedge v_2\} \cup \{v_1 \wedge e_{i2} \wedge v_2\}$ (Figure 7.3(b)).



(a) $e_i = e_{i1} \wedge e_{i2}$ (b) $e_i = e_{i1} \vee e_{i2}$

Figure 7.3 Application of associative and distributive laws.

Post-process Step:

When the Process Step is finished, we have $V = \Phi$ and

$$U = \{\overline{KB}_1, \overline{KB}_2, \cdots, \overline{KB}_{\bar{\kappa}}\},$$

which is equivalent to:

$$KB = \overline{KB}_1 \vee \overline{KB}_2 \vee \cdots \vee \overline{KB}_{\bar{\kappa}}.$$

Each knowledge base $\overline{KB}_i = \bar{e}_{i1} \wedge \bar{e}_{i2} \wedge \cdots \wedge \bar{e}_{i\gamma_i}$ $(i = 1, \cdots, \bar{\kappa})$ is primitive. Process $\overline{KB}_i$ as follows:

- If $\overline{KB}_i$ has two identical logical expressions, reduce them into one;
- If $\overline{KB}_i$ has two conflicting logical expressions, i.e., $\bar{e}_{ij_1} = p$ and $\bar{e}_{ij_2} = \neg p$ $(j_1 \neq j_2)$, let $U = U - \{\overline{KB}_i\}$.

Finally we have:

$$U = \{KB_1, KB_2, \cdots, KB_\kappa\},\ \kappa \leq \bar{\kappa},$$

which is equivalent to:

$$KB = KB_1 \vee KB_2 \vee \cdots \vee KB_\kappa .$$

Each knowledge base KB_i ($i = 1, \cdots, \kappa$) is in the form of:

$$KB_i = e_{i1} \wedge e_{i2} \wedge \cdots \wedge e_{i\mathrm{M}} ,$$

where M is the number of processes in Ω. Each KB_i corresponds to a unique life locus LL_i which is defined as:

$$LL_i(p) = \begin{cases} \text{True} & KB_i = \cdots \wedge p \wedge \cdots \\ \text{False} & KB_i = \cdots \wedge \neg p \wedge \cdots \end{cases} \quad \forall p \in \Omega .$$

7.2.3 Algorithm Analysis

We can prove the correctness, completeness, and complexity of the logical deduction algorithm as follows.

Theorem 1 (Correctness): Every life locus generated by the logical deduction algorithm satisfies KB.

Proof:

Using associative and distributive laws, KB has been transformed equivalently by the logical deduction algorithm to:

$$KB = KB_1 \vee KB_2 \vee \cdots \vee KB_\kappa .$$

Each KB_i generates a unique life locus LL_i, i.e., LL_i is a unique life locus that satisfies KB_i. From the definition of logical disjunction, LL_i also satisfies KB. This proves the correctness of the logical deduction algorithm, i.e., every life locus generated by the algorithm is correct.

Theorem 2 (Completeness): Every life locus that satisfies KB can be generated by the logical deduction algorithm.

Proof:

Given $KB = KB_1 \vee KB_2 \vee \cdots \vee KB_\kappa$, the logical deduction algorithm generates κ life loci, i.e., resulting in a solution set:

$$\Sigma = \{LL_1, LL_2, \cdots, LL_\kappa\},$$

where LL_i is a unique life locus that satisfies KB_i. Suppose that there exists a $(\kappa+1)$th life locus $LL_{\kappa+1} \notin \Sigma$ but satisfies KB. From the unique correspondence between KB_i and LL_i $(i = 1, \cdots, \kappa)$, KB_i $(i = 1, \cdots, \kappa)$ is False for $LL_{\kappa+1}$. From the definition of logical disjunction, $LL_{\kappa+1}$ also bears a False value for KB, i.e., does not satisfy KB. This proves the completeness of the solution set Σ of the logical deduction algorithm.

Theorem 3 (Complexity): The worst-case time complexity of the logical deduction algorithm is $O(2^N)$, where N is the number of disjunction operators in KB.

Proof:

Given a standard knowledge base KB that contains N disjunction operators. From the Process Step of the logical deduction algorithm, each disjunction operator results in an operation of splitting the knowledge base into two knowledge bases (Figure 7.3(b)). These splitting operations consume most of the algorithm execution time.

Let $S(N)$ represent the worst-case total number of splitting operations needed. We have (in the worst case):

$$\begin{array}{ccc} S(N) & = & 1+2S(N-1), \\ S(N-1) & = & 1+2S(N-2), \\ \vdots & & \vdots \\ S(1) & = & 1, \end{array}$$

and finally:

$$S(N) = 1 + 2^1 + 2^2 + \cdots + 2^{N-1} = 2^N - 1 .$$

Suppose that each splitting operation requires T_s units of time, the logical deduction algorithm then has the following worst-case time complexity $T(N)$:

$$T(N) = (2^N - 1)T_s = O(2^N) .$$

7.3 Algorithm Implementation

Based on the logical deduction algorithm, we can provide a procedure by which the algorithm can be implemented efficiently in computer programs. Let Σ represent a solution set. Each solution in Σ has four components, i.e.,

$$(K\hat{B}, l \mid K\hat{B}, LL \mid K\hat{B}, t \mid K\hat{B}) .$$

$K\hat{B}$ is a knowledge base, $l \mid K\hat{B}$ is a location identifier pointing to the next unprocessed knowledge item in $K\hat{B}$, $LL \mid K\hat{B}$ is a corresponding life locus, and $t \mid K\hat{B}$ is a property tag denoting whether this solution has been completely processed ($t \mid K\hat{B} = 1$) or not ($t \mid K\hat{B} = 0$). For $\forall p \in \Omega$, $LL(p) \mid K\hat{B}$ may have three different values: True, False, and Unknown, where Unknown represents an intermediate decision status during algorithm execution.

The implementation procedure of the logical deduction algorithm is as follows:

Step1. Initialize $\Sigma = \{(KB, l \mid KB, LL \mid KB, t \mid KB)\}$, where KB is a knowledge base provided by product developers, $l \mid KB = 1$ points to the first knowledge item in KB, $LL(p) \mid KB = \text{Unknown}$ ($\forall p \in \Omega$), and $t \mid KB = 0$.

Step 2. Select an uncompleted solution from Σ and denote it as the current solution:

$$(KB_c, l \mid KB_c, LL \mid KB_c, t \mid KB_c).$$

Step 3. Process $(KB_c, l \mid KB_c, LL \mid KB_c, t \mid KB_c)$ as follows:

(3.1) Obtain the next unprocessed knowledge item $e = e_{l \mid KB_c}$:

- If $e = p$, check the value of $LL(p) \mid KB_c$:
    - If $LL(p) \mid KB_c$ = Unknown, let $LL(p) \mid KB_c$ = True, increment $l \mid KB_c$ by 1.
    - If $LL(p) \mid KB_c$ = True, increment $l \mid KB_c$ by 1.
    - If $LL(p) \mid KB_c$ = False, remove $(KB_c, l \mid KB_c, LL \mid KB_c, t \mid KB_c)$ from Σ and go to Step 4.
- If $e = \neg p$, check the value of $LL(p) \mid KB_c$:
    - If $LL(p) \mid KB_c$ = Unknown, let $LL(p) \mid KB_c$ = False, increment $l \mid KB_c$ by 1.
    - If $LL(p) \mid KB_c$ = False, increment $l \mid KB_c$ by 1.
    - If $LL(p) \mid KB_c$ = True, remove $(KB_c, l \mid KB_c, LL \mid KB_c, t \mid KB_c)$ from Σ and go to Step 4.
- If $e = e_1 \wedge e_2$, modify $(KB_c, l \mid KB_c, LL \mid KB_c, t \mid KB_c)$ as follows:
    - In KB_c, substitute e with e_1, and insert e_2 after e_1;
    - Keep $l \mid KB_c$, $LL \mid KB_c$, and $t \mid KB_c$ unchanged.
- If $e = e_1 \vee e_2$, first modify $(KB_c, l \mid KB_c, LL \mid KB_c, t \mid KB_c)$ as follows:
    - In KB_c, substitute e with e_1;
    - Keep $l \mid KB_c$, $LL \mid KB_c$, and $t \mid KB_c$ unchanged.

  Then generate a new solution $(KB_{\text{new}}, l \mid KB_{\text{new}}, LL \mid KB_{\text{new}}, t \mid KB_{\text{new}})$ and insert it into Σ. It is initialized as follows:

- KB_{new} is the same as KB_c after modification except that e_1 is substituted with e_2;
- $l \mid KB_{\text{new}} = l \mid KB_c$, $LL \mid KB_{\text{new}} = LL \mid KB_c$, and $t \mid KB_{\text{new}} = 0$.

(3.2) If the last knowledge item in KB_c has been processed, let $t \mid KB_c = 1$ and go to Step 4; Otherwise continue with Step (3.1) and process the next unprocessed knowledge item in KB_c.

Step 4. If all the solutions in Σ are tagged with 1, finish. Otherwise continue with Step 2 and process the next uncompleted solution.

When this procedure is finished, every solution $(KB_i, l \mid KB_i, LL \mid KB_i, t \mid KB_i)$ in Σ corresponds to a unique life locus that is recorded in $LL \mid KB_i$ $(i = 1, \cdots, \kappa)$. The procedure can be implemented along with a depth-first search algorithm that searches for an optimal life locus simultaneously during the automatic life locus setup.

7.4 Application Examples

In this section, we apply the proposed methodology to two product development examples: cup and personal computer (PC). Cup has a simple ELCS and PC has a complex one.

7.4.1 Example 1: Cup Development

Consider the development of a cup. Suppose that product developers expect a cup to have an ELCS as a sequence of four consecutive life phases: (1) Design, (2) Manufacture, (3) Use, and (4) Recycle. The possible processes in each life phase are identified as in Table 7.1 (the same as Table 3.1, reproduced here for convenience).

Table 7.1 Possible processes in a cup's life cycle.

Life Phase	Process	Meaning
Design	D1	Foam cup design
	D2	Plastic cup design
	D3	Glass cup design
Manufacture	M1	Manufacture foam cup
	M2	Manufacture plastic cup
	M3	Manufacture glass cup
Use	U	Use the cup
Recycle	R1	Foam cup recycled for secondary use (e.g. packaging)
	R2	Plastic cup material recycling process A
	R3	Plastic cup material recycling process B
	R4	Glass cup material recycling process C
	R5	Glass cup material recycling process D

The overall process set $\Omega = \{D1, D2, D3, M1, M2, M3, U, R1, R2, R3, R4, R5\}$. The logical relations among processes in Ω are identified as in Table 7.2.

Table 7.2 Knowledge base *KB* for a cup.

Knowledge Item	Meaning
$D1 \uparrow D2 \uparrow D3$	Select a design
$M1 \uparrow M2 \uparrow M3$	Select a manufacturing process
$D1 \rightarrow M1$	If D1 is selected, select M1
$D2 \rightarrow M2$	If D2 is selected, select M2
$D3 \rightarrow M3$	If D3 is selected, select M3
U	Select use process
$R1 \uparrow R2 \uparrow R3 \uparrow R4 \uparrow R5$	Select a recycling process
$D1 \rightarrow R1$	If D1 is selected, select R1
$D2 \rightarrow (R2 \uparrow R3)$	If D2 is selected, select R2 or R3
$D3 \rightarrow (R4 \uparrow R5)$	If D3 is selected, select R4 or R5

After standardization, we obtain a standard knowledge base *KB* as shown in Table 7.3. Applying the logical deduction algorithm to this standard knowledge base, we obtain all the feasible life loci for a cup as shown in Table 7.4. Each life locus shows only its selected processes.

Table 7.3 Standard knowledge base *KB* for a cup.

$(D1 \wedge \neg D2 \wedge \neg D3) \vee (\neg D1 \wedge D2 \wedge \neg D3) \vee (\neg D1 \wedge \neg D2 \wedge D3)$
$(M1 \wedge \neg M2 \wedge \neg M3) \vee (\neg M1 \wedge M2 \wedge \neg M3) \vee (\neg M1 \wedge \neg M2 \wedge M3)$
$\neg D1 \vee M1$
$\neg D2 \vee M2$
$\neg D3 \vee M3$
U
$(R1 \wedge \neg R2 \wedge \neg R3 \wedge \neg R4 \wedge \neg R5) \vee (\neg R1 \wedge R2 \wedge \neg R3 \wedge \neg R4 \wedge \neg R5)$ $\vee (\neg R1 \wedge \neg R2 \wedge R3 \wedge \neg R4 \wedge \neg R5) \vee (\neg R1 \wedge \neg R2 \wedge \neg R3 \wedge R4 \wedge \neg R5)$ $\vee (\neg R1 \wedge \neg R2 \wedge \neg R3 \wedge \neg R4 \wedge R5)$
$\neg D1 \vee R1$
$\neg D2 \vee ((\neg R2 \wedge R3) \vee (R2 \wedge \neg R3))$
$\neg D3 \vee ((\neg R4 \wedge R5) \vee (R4 \wedge \neg R5))$

Table 7.4 Feasible life loci for a cup.

LL_1	LL_2	LL_3	LL_4	LL_5
D1 M1 U R1	D2 M2 U R2	D2 M2 U R3	D3 M3 U R4	D3 M3 U R5

7.4.2 Example 2: PC Development

Consider the development of a PC. Figure 7.4 illustrates a PC's ELCS (the same as Figure 5.1, reproduced here for convenience). It has eight life phases: (1) Architecture Design, (2) Component Selection, (3) Assembly & Testing, (4) Use, (5) Upgrade, (6) Disassembly, (7) Component Refurbish, and (8) Component Landfill.

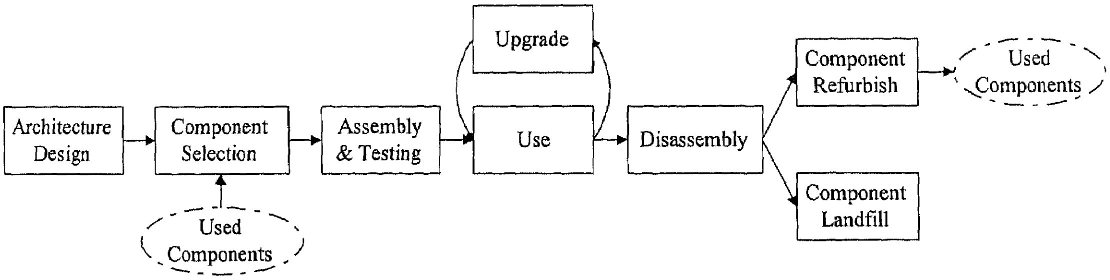


Figure 7.4 An ELCS for PC.

The key processes in each life phase are identified as in Table 7.5 (referring to Tables 5.1 and 5.3).

Table 7.5 Key processes in a PC's life cycle.

Life Phase	Process	Meaning
Architecture Design	D1	Conventional low cost PC mini-tower
	D2	Mini-tower PC by Compaq
Component Selection	MON1	Used 15" monitor
	MON2	15" monitor
	MON3	17" monitor
	CPU1	Used 200MHz CPU
	CPU2	200MHz CPU
	CPU3	233MHz CPU
	MEM1	Used 16MB memory
	MEM2	16MB memory
	MEM3	32MB memory
	SB1	TX QDI system board
	SB2	VX QDI Jumpless system board
	HD1	Used 3.1G hard disk
	HD2	3.1G hard disk
	HD3	5.1G hard disk
	CASE1	Used D1 case
	CASE2	D1 case
	CASE3	Used D2 case
	CASE4	D2 case
	EMS	Energy Management Software
Assembly & Testing	ASM1	Assembly sequence of D1
	ASM2	Assembly sequence of D2
Use	USEMON1	Use MON1
	USEMON2	Use MON2
	USEMON3	Use MON3
	USECPU1	Use CPU1
	USECPU2	Use CPU2
	USECPU3	Use CPU3
	USEMEM1	Use MEM1
	USEMEM2	Use MEM2
	USEMEM3	Use MEM3
	USESB1	Use SB1
	USESB2	Use SB2
	USEHD1	Use HD1
	USEHD2	Use HD2
	USEHD3	Use HD3
	USEEMS	Use EMS
Upgrade	UPGMON1	Upgrade to 17" or 21"
	UPGMON2	Upgrade to 17" or 21"
	UPGMON3	Upgrade to 21"
	UPGCPU1	Upgrade to 233MHz or 266MHz
	UPGCPU2	Upgrade to 233MHz or 266MHz
	UPGCPU3	Upgrade to 266MHz
	UPGMEM1	Upgrade to 32MB or 64MB
	UPGMEM2	Upgrade to 32MB or 64MB
	UPGMEM3	Upgrade to 64MB
	UPGHD1	Upgrade to 4.0G or 5.1 G
	UPGHD2	Upgrade to 4.0G or 5.1 G
	UPGHD3	Upgrade to 6.5G
	UPGEMS	Upgrade to EMS

Table 7.5 (continued)

Life Phase	Process	Meaning
Disassembly	DISASM1	Disassembly sequence of D1
	DISASM2	Disassembly sequence of D2
Component Refurbish	REFMON1	Refurbish MON1
	REFMON2	Refurbish MON2
	REFMON3	Refurbish MON3
	REFCPU1	Refurbish CPU1
	REFCPU2	Refurbish CPU2
	REFCPU3	Refurbish CPU3
	REFMEM1	Refurbish MEM1
	REFMEM2	Refurbish MEM2
	REFMEM3	Refurbish MEM3
	REFHD1	Refurbish HD1
	REFHD2	Refurbish HD2
	REFHD3	Refurbish HD3
	REFCASE1	Refurbish CASE1
	REFCASE2	Refurbish CASE2
	REFCASE3	Refurbish CASE3
	REFCASE4	Refurbish CASE4
Component Landfill	LFMON1	Landfill MON1
	LFMON2	Landfill MON2
	LFMON3	Landfill MON3
	LFCPU1	Landfill CPU1
	LFCPU2	Landfill CPU2
	LFCPU3	Landfill CPU3
	LFMEM1	Landfill MEM1
	LFMEM2	Landfill MEM2
	LFMEM3	Landfill MEM3
	LFHD1	Landfill HD1
	LFHD2	Landfill HD2
	LFHD3	Landfill HD3
	LFCASE1	Landfill CASE1
	LFCASE2	Landfill CASE2
	LFCASE3	Landfill CASE3
	LFCASE4	Landfill CASE4

The logical relations among these processes are identified as in Table 7.6.

Table 7.6 Knowledge base *KB* for a PC.

Knowledge Item	Meaning
D1 ↑ D2	Select a design
MON1 ↑ MON2 ↑ MON3	Select a monitor
CPU1 ↑ CPU2 ↑ CPU3	Select a CPU
MEM1 ↑ MEM2 ↑ MEM3	Select a memory
SB1 ↑ SB2	Select a system board
HD1 ↑ HD2 ↑ HD3	Select a hard disk
CASE1 ↑ CASE2 ↑ CASE3 ↑ CASE4	Select a case
EMS ↑ ¬ EMS	Select EMS or not

Table 7.6 (continued)

Knowledge Item	Meaning
D1 → (CASE1 ↑ CASE2)	If D1 is selected, select CASE1 or CASE2
D2 → (CASE3 ↑ CASE4)	If D2 is selected, select CASE3 or CASE4
ASM1 ↑ ASM2	Select an assembly sequence
DISASM1 ↑ DISASM2	Select a disassembly sequence
D1 → (ASM1 ∧ DISASM1)	If D1 is selected, select ASM1 and DISASM1
D2 → (ASM2 ∧ DISASM2)	If D2 is selected, select ASM2 and DISASM2
USEMON1 ↑ USEMON2 ↑ USEMON3	Select a use process for monitor
UPGMON1 ↑ UPGMON2 ↑ UPGMON3	Select an upgrade process for monitor
REFMON1 ↑ REFMON2 ↑ REFMON3	Select a refurbish process for monitor
LFMON1 ↑ LFMON2 ↑ LFMON3	Select a landfill process for monitor
USECPU1 ↑ USECPU2 ↑ USECPU3	Select a use process for CPU
UPGCPU1 ↑ UPGCPU2 ↑ UPGCPU3	Select an upgrade process for CPU
REFCPU1 ↑ REFCPU2 ↑ REFCPU3	Select a refurbish process for CPU
LFCPU1 ↑ LFCPU2 ↑ LFCPU3	Select a landfill process for CPU
USEMEM1 ↑ USEMEM2 ↑ USEMEM3	Select a use process for memory
UPGMEM1 ↑ UPGMEM2 ↑ UPGMEM3	Select an upgrade process for memory
REFMEM1 ↑ REFMEM2 ↑ REFMEM3	Select a refurbish process for memory
LFMEM1 ↑ LFMEM2 ↑ LFMEM3	Select a landfill process for memory
USEHD1 ↑ USEHD2 ↑ USEHD3	Select a use process for hard disk
UPGHD1 ↑ UPGHD2 ↑ UPGHD3	Select an upgrade process for hard disk
REFHD1 ↑ REFHD2 ↑ REFHD3	Select a refurbish process for hard disk
LFHD1 ↑ LFHD2 ↑ LFHD3	Select a landfill process for hard disk
MON1 → (USEMON1 ∧ UPGMON1 ∧ REFMON1 ∧ LFMON1)	If MON1 is selected, select USEMON1, UPGMON1, REFMON1, and LFMON1
MON2 → (USEMON2 ∧ UPGMON2 ∧ REFMON2 ∧ LFMON2)	Similar to the above explanation
MON3 → (USEMON3 ∧ UPGMON3 ∧ REFMON3 ∧ LFMON3)	Similar to the above explanation
CPU1 → (USECPU1 ∧ UPGCPU1 ∧ REFCPU1 ∧ LFCPU1)	Similar to the above explanation
CPU2 → (USECPU2 ∧ UPGCPU2 ∧ REFCPU2 ∧ LFCPU2)	Similar to the above explanation
CPU3 → (USECPU3 ∧ UPGCPU3 ∧ REFCPU3 ∧ LFCPU3)	Similar to the above explanation
MEM1 → (USEMEM1 ∧ UPGMEM1 ∧ REFMEM1 ∧ LFMEM1)	Similar to the above explanation
MEM2 → (USEMEM2 ∧ UPGMEM2 ∧ REFMEM2 ∧ LFMEM2)	Similar to the above explanation
MEM3 → (USEMEM3 ∧ UPGMEM3 ∧ REFMEM3 ∧ LFMEM3)	Similar to the above explanation
HD1 → (USEHD1 ∧ UPGHD1 ∧ REFHD1 ∧ LFHD1)	Similar to the above explanation
HD2 → (USEHD2 ∧ UPGHD2 ∧ REFHD2 ∧ LFHD2)	Similar to the above explanation
HD3 → (USEHD3 ∧ UPGHD3 ∧ REFHD3 ∧ LFHD3)	Similar to the above explanation
USESB1 ↑ USESB2	Select a use process for system board
SB1 → USESB1	If SB1 is selected, select USESB1
SB2 → USESB2	If SB2 is selected, select USESB2
USEEMS ↑ ¬ USEEMS	Use EMS or not
UPGEMS ↑ ¬ UPGEMS	Upgrade to EMS or not

Table 7.6 (continued)

Knowledge Item	Meaning
$EMS \rightarrow USEEMS$	If EMS is selected, select USEEMS
$\neg EMS \rightarrow \neg USEEMS$	If EMS is not selected, do not select USEEMS
$EMS \rightarrow \neg UPGEMS$	If EMS is selected, do not select UPGEMS
$\neg EMS \rightarrow UPGEMS$	If EMS is not selected, select UPGEMS
$REFCASE1 \uparrow REFCASE2 \uparrow REFCASE3 \uparrow REFCASE4$	Select a refurbish process for case
$LFCASE1 \uparrow LFCASE2 \uparrow LFCASE3 \uparrow LFCASE4$	Select a landfill process for case
$CASE1 \rightarrow (REFCASE1 \wedge LFCASE1)$	If CASE1 is selected, select REFCASE1 and LFCASE1
$CASE2 \rightarrow (REFCASE2 \wedge LFCASE2)$	Similar to the above explanation
$CASE3 \rightarrow (REFCASE3 \wedge LFCASE3)$	Similar to the above explanation
$CASE4 \rightarrow (REFCASE4 \wedge LFCASE4)$	Similar to the above explanation

After standardization, we obtain a standard knowledge base *KB* as shown in Table 7.7.

Table 7.7 Standard knowledge base *KB* for a PC.

$(D1 \wedge \neg D2) \vee (\neg D1 \wedge D2)$
$(MON1 \wedge \neg MON2 \wedge \neg MON3) \vee (\neg MON1 \wedge MON2 \wedge \neg MON3) \vee (\neg MON1 \wedge \neg MON2 \wedge MON3)$
$(CPU1 \wedge \neg CPU2 \wedge \neg CPU3) \vee (\neg CPU1 \wedge CPU2 \wedge \neg CPU3) \vee (\neg CPU1 \wedge \neg CPU2 \wedge CPU3)$
$(MEM1 \wedge \neg MEM2 \wedge \neg MEM3) \vee (\neg MEM1 \wedge MEM2 \wedge \neg MEM3) \vee (\neg MEM1 \wedge \neg MEM2 \wedge MEM3)$
$(SB1 \wedge \neg SB2) \vee (\neg SB1 \wedge SB2)$
$(HD1 \wedge \neg HD2 \wedge \neg HD3) \vee (\neg HD1 \wedge HD2 \wedge \neg HD3) \vee (\neg HD1 \wedge \neg HD2 \wedge HD3)$
$(CASE1 \wedge \neg CASE2 \wedge \neg CASE3 \wedge \neg CASE4) \vee (\neg CASE1 \wedge CASE2 \wedge \neg CASE3 \wedge \neg CASE4) \vee (\neg CASE1 \wedge \neg CASE2 \wedge CASE3 \wedge \neg CASE4) \vee (\neg CASE1 \wedge \neg CASE2 \wedge \neg CASE3 \wedge CASE4)$
$EMS \vee \neg EMS$
$\neg D1 \vee ((CASE1 \wedge \neg CASE2) \vee (\neg CASE1 \wedge CASE2))$
$\neg D2 \vee ((CASE3 \wedge \neg CASE4) \vee (\neg CASE3 \wedge CASE4))$
$(ASM1 \wedge \neg ASM2) \vee (\neg ASM1 \wedge ASM2)$
$(DISASM1 \wedge \neg DISASM2) \vee (\neg DISASM1 \wedge DISASM2)$
$\neg D1 \vee (ASM1 \wedge DISASM1)$
$\neg D2 \vee (ASM2 \wedge DISASM2)$
$(USEMON1 \wedge \neg USEMON2 \wedge \neg USEMON3) \vee (\neg USEMON1 \wedge USEMON2 \wedge \neg USEMON3) \vee (\neg USEMON1 \wedge \neg USEMON2 \wedge USEMON3)$
$(UPGMON1 \wedge \neg UPGMON2 \wedge \neg UPGMON3) \vee (\neg UPGMON1 \wedge UPGMON2 \wedge \neg UPGMON3) \vee (\neg UPGMON1 \wedge \neg UPGMON2 \wedge UPGMON3)$
$(REFMON1 \wedge \neg REFMON2 \wedge \neg REFMON3) \vee (\neg REFMON1 \wedge REFMON2 \wedge \neg REFMON3) \vee (\neg REFMON1 \wedge \neg REFMON2 \wedge REFMON3)$
$(LFMON1 \wedge \neg LFMON2 \wedge \neg LFMON3) \vee (\neg LFMON1 \wedge LFMON2 \wedge \neg LFMON3) \vee (\neg LFMON1 \wedge \neg LFMON2 \wedge LFMON3)$
$(USECPU1 \wedge \neg USECPU2 \wedge \neg USECPU3) \vee (\neg USECPU1 \wedge USECPU2 \wedge \neg USECPU3) \vee (\neg USECPU1 \wedge \neg USECPU2 \wedge USECPU3)$

Table 7.7 (continued)

$(UPGCPU1 \land \neg UPGCPU2 \land \neg UPGCPU3) \lor (\neg UPGCPU1 \land UPGCPU2 \land \neg UPGCPU3) \lor (\neg UPGCPU1 \land \neg UPGCPU2 \land UPGCPU3)$
$(REFCPU1 \land \neg REFCPU2 \land \neg REFCPU3) \lor (\neg REFCPU1 \land REFCPU2 \land \neg REFCPU3) \lor (\neg REFCPU1 \land \neg REFCPU2 \land REFCPU3)$
$(LFCPU1 \land \neg LFCPU2 \land \neg LFCPU3) \lor (\neg LFCPU1 \land LFCPU2 \land \neg LFCPU3) \lor (\neg LFCPU1 \land \neg LFCPU2 \land LFCPU3)$
$(USEMEM1 \land \neg USEMEM2 \land \neg USEMEM3) \lor (\neg USEMEM1 \land USEMEM2 \land \neg USEMEM3) \lor (\neg USEMEM1 \land \neg USEMEM2 \land USEMEM3)$
$(UPGMEM1 \land \neg UPGMEM2 \land \neg UPGMEM3) \lor (\neg UPGMEM1 \land UPGMEM2 \land \neg UPGMEM3) \lor (\neg UPGMEM1 \land \neg UPGMEM2 \land UPGMEM3)$
$(REFMEM1 \land \neg REFMEM2 \land \neg REFMEM3) \lor (\neg REFMEM1 \land REFMEM2 \land \neg REFMEM3) \lor (\neg REFMEM1 \land \neg REFMEM2 \land REFMEM3)$
$(LFMEM1 \land \neg LFMEM2 \land \neg LFMEM3) \lor (\neg LFMEM1 \land LFMEM2 \land \neg LFMEM3) \lor (\neg LFMEM1 \land \neg LFMEM2 \land LFMEM3)$
$(USEHD1 \land \neg USEHD2 \land \neg USEHD3) \lor (\neg USEHD1 \land USEHD2 \land \neg USEHD3) \lor (\neg USEHD1 \land \neg USEHD2 \land USEHD3)$
$(UPGHD1 \land \neg UPGHD2 \land \neg UPGHD3) \lor (\neg UPGHD1 \land UPGHD2 \land \neg UPGHD3) \lor (\neg UPGHD1 \land \neg UPGHD2 \land UPGHD3)$
$(REFHD1 \land \neg REFHD2 \land \neg REFHD3) \lor (\neg REFHD1 \land REFHD2 \land \neg REFHD3) \lor (\neg REFHD1 \land \neg REFHD2 \land REFHD3)$
$(LFHD1 \land \neg LFHD2 \land \neg LFHD3) \lor (\neg LFHD1 \land LFHD2 \land \neg LFHD3) \lor (\neg LFHD1 \land \neg LFHD2 \land LFHD3)$
$\neg MON1 \lor (USEMON1 \land UPGMON1 \land REFMON1 \land LFMON1)$
$\neg MON2 \lor (USEMON2 \land UPGMON2 \land REFMON2 \land LFMON2)$
$\neg MON3 \lor (USEMON3 \land UPGMON3 \land REFMON3 \land LFMON3)$
$\neg CPU1 \lor (USECPU1 \land UPGCPU1 \land REFCPU1 \land LFCPU1)$
$\neg CPU2 \lor (USECPU2 \land UPGCPU2 \land REFCPU2 \land LFCPU2)$
$\neg CPU3 \lor (USECPU3 \land UPGCPU3 \land REFCPU3 \land LFCPU3)$
$\neg MEM1 \lor (USEMEM1 \land UPGMEM1 \land REFMEM1 \land LFMEM1)$
$\neg MEM2 \lor (USEMEM2 \land UPGMEM2 \land REFMEM2 \land LFMEM2)$
$\neg MEM3 \lor (USEMEM3 \land UPGMEM3 \land REFMEM3 \land LFMEM3)$
$\neg HD1 \lor (USEHD1 \land UPGHD1 \land REFHD1 \land LFHD1)$
$\neg HD2 \lor (USEHD2 \land UPGHD2 \land REFHD2 \land LFHD2)$
$\neg HD3 \lor (USEHD3 \land UPGHD3 \land REFHD3 \land LFHD3)$
$(USESB1 \land \neg USESB2) \lor (\neg USESB1 \land USESB2)$
$\neg SB1 \lor USESB1$
$\neg SB2 \lor USESB2$
$USEEMS \lor \neg USEEMS$
$UPGEMS \lor \neg UPGEMS$
$\neg EMS \lor USEEMS$
$EMS \lor \neg USEEMS$
$\neg EMS \lor \neg UPGEMS$
$EMS \lor UPGEMS$

Table 7.7 (continued)

$(\mathrm{REFCASE1} \wedge \neg \mathrm{REFCASE2} \wedge \neg \mathrm{REFCASE3} \wedge \neg \mathrm{REFCASE4})$ $\vee (\neg \mathrm{REFCASE1} \wedge \mathrm{REFCASE2} \wedge \neg \mathrm{REFCASE3} \wedge \neg \mathrm{REFCASE4})$ $\vee (\neg \mathrm{REFCASE1} \wedge \neg \mathrm{REFCASE2} \wedge \mathrm{REFCASE3} \wedge \neg \mathrm{REFCASE4})$ $\vee (\neg \mathrm{REFCASE1} \wedge \neg \mathrm{REFCASE2} \wedge \neg \mathrm{REFCASE3} \wedge \mathrm{REFCASE4})$
$(\mathrm{LFCASE1} \wedge \neg \mathrm{LFCASE2} \wedge \neg \mathrm{LFCASE3} \wedge \neg \mathrm{LFCASE4})$ $\vee (\neg \mathrm{LFCASE1} \wedge \mathrm{LFCASE2} \wedge \neg \mathrm{LFCASE3} \wedge \neg \mathrm{LFCASE4})$ $\vee (\neg \mathrm{LFCASE1} \wedge \neg \mathrm{LFCASE2} \wedge \mathrm{LFCASE3} \wedge \neg \mathrm{LFCASE4})$ $\vee (\neg \mathrm{LFCASE1} \wedge \neg \mathrm{LFCASE2} \wedge \neg \mathrm{LFCASE3} \wedge \mathrm{LFCASE4})$
$\neg \mathrm{CASE1} \vee (\mathrm{REFCASE1} \wedge \mathrm{LFCASE1})$
$\neg \mathrm{CASE2} \vee (\mathrm{REFCASE2} \wedge \mathrm{LFCASE2})$
$\neg \mathrm{CASE3} \vee (\mathrm{REFCASE3} \wedge \mathrm{LFCASE3})$
$\neg \mathrm{CASE4} \vee (\mathrm{REFCASE4} \wedge \mathrm{LFCASE4})$

Applying the logical deduction algorithm to this standard knowledge base, we can obtain 1296 life loci for a PC. Due to space limit, they are not listed here.

7.5 Summary

Relations among processes in a product's life cycle are an important part of product developers' knowledge and expertise. In our previous approach, product developers not only have to consider the process relations, but also need to provide manually all the feasible combinations of processes for further optimization. This procedure is not trivial and its accuracy may not be guaranteed when a large problem is encountered. To overcome this barrier to design automation of competitive and environmentally conscious products and processes, this chapter formally describes relations among processes and all related rules using logic. A logical deduction algorithm is used to generate automatically all the feasible life loci for a target product. Properties and complexity of the algorithm are also analyzed. Using this approach, product developers need only consider all the logical relations among processes in a product's life cycle and the rest is automatically performed. This is demonstrated through cup and PC development examples. The logical deduction algorithm can be implemented along with a depth-first search algorithm that searches for an optimal life locus simultaneously during the automatic life locus setup.

CHAPTER 8

A LIFE CYCLE ENGINEERING APPROACH TO FMS DEVELOPMENT

The previous chapters focus on "product" development. However, the same IPPD methodology can be applied to the life cycle development of manufacturing systems as well. Taking a manufacturing system as a complex product [Talavage and Hannam, 1988; Cantamessa, 1998], we can similarly model and solve the various optimization problems associated with its development. This chapter takes a flexible manufacturing system (FMS) as an example, and applies the IPPD methodology to its development.

FMSs have emerged in recent years as a viable answer to meet the market demand for increased product variety, short product life cycles, and uncertain demand [Rajasekharan, Peters, and Yang, 1998]. However, implementing an FMS is very costly, and the investment tends to be irreversible, thus necessarily requiring careful consideration before a decision is made [ElMaraghy and Ravi, 1992; Myint and Tabucanon, 1994].

An FMS design problem consists of three sub-problems: machine selection problem, facility layout problem (FLP), and machine layout problem (MLP). The FLP and MLP have been studied by numerous researchers [Heragu and Kusiak, 1988; Das, 1993; Rajasekharan, Peters, and Yang, 1998] and are not considered here. This chapter focuses on using the IPPD methodology to solve the FMS machine selection problem, and thus providing a life cycle engineering approach to this typical FMS design problem that is usually addressed by such traditional approaches as analytic hierarchy process and goal programming [Myint and Tabucanon, 1994].

As stated earlier, FMS implementation requires a huge amount of capital investment. Therefore when a particular FMS is selected, the company expects it to stay competitive for a relatively long period of time. In our research effort of applying the IPPD methodology to FMS development, we find that for such expensive and long-lasting systems, more modeling capability and higher decision accuracy are needed. For example, if the technology of a particular FMS component develops quickly, it may lead to the inability of the company to adjust as rapidly as would be possible with a subsequent advanced equipment. Since investment in flexible technologies is usually large, the obsolescence potential requires careful consideration at the time of selection [Abdel-Malek and Wolf, 1994].

To cope with this problem, this chapter introduces time variable into the IPPD methodology. The execution duration of processes and their time-varying characteristics are considered to increase the modeling capability and decision accuracy of the methodology.

This chapter is organized as follows. Section 8.1 introduces a timed IPPD methodology with its new concepts and solution method. Section 8.2 applies the methodology to the life cycle development of FMSs.

8.1 A Timed IPPD Methodology

In this section, we introduce a timed IPPD methodology. Time-related new concepts are defined in Section 8.1.1. A timed generic framework is introduced in Section 8.1.2 to describe the timed IPPD problem, and a timed solution methodology is provided in Section 8.1.3.

8.1.1 Time-related Concepts

In the timed IPPD methodology, the following concepts are defined: timed process, index function vector, time-invariance, and classification of timed processes.

(1) Timed Process

In the timed IPPD methodology, we consider the execution duration of a process and represent a process as:

$$p(t_p, t_p^0),\ t_p^0 \le t_p \le t_p^1,$$

where t_p^0 is process p 's execution starting time, t_p^1 is its execution ending time, t_p is its present time, and $t_p^1 - t_p^0$ is its execution time.

(2) Index Function Vector

An index function of process p is a time function $c_p(t_p, t_p^0)$ representing the cumulative value of an interesting characteristic of the process evaluated at t_p. α index functions form an index function vector:

$$\mathbf{c}_p(t_p, t_p^0) = \begin{pmatrix} c_{1p}(t_p, t_p^0) \\ \vdots \\ c_{ip}(t_p, t_p^0) \\ \vdots \\ c_{\alpha p}(t_p, t_p^0) \end{pmatrix},\ t_p^0 \le t_p \le t_p^1.$$

Generally speaking, process p 's index function vector $\mathbf{c}_p(t_p, t_p^0)$ depends on both t_p and t_p^0.

(3) Time-invariance

A process $p(t_p, t_p^0)$ is called time-invariant if:

$$\mathbf{c}_p(t_p, t_p^0) = \mathbf{c}_p(t_p - t_p^0),\ t_p^0 \le t_p \le t_p^1,$$

i.e., its index function vector depends only on the difference between t_p and t_p^0.

In the timed IPPD methodology, we assume that all the processes can be considered as time-invariant with sufficient modeling capability. With this assumption, a process can be represented as $p(t_p - t_p^0)$ and its index function vector as $\mathbf{c}_p(t_p - t_p^0)$.

(4) Classification of Timed Processes

A timed process is classified into two types: fixed execution time process and varying execution time process.

- $p(t_p - t_p^0)$ is a fixed execution time process if its execution time $t_p^1 - t_p^0$ is fixed;
- $p(t_p - t_p^0)$ is a varying execution time process if its execution time $t_p^1 - t_p^0$ is not fixed and needs to be determined by product developers.

Take an example of a product's manufacturing process and use process. The time of its manufacturing process is relatively fixed compared with its use process. Therefore the former is often regarded as a fixed execution time process and the latter as a varying execution time process.

8.1.2 Timed Generic Framework

Based on these time-related concepts, we can formulate a timed generic framework to describe the timed IPPD problem.

Represent the target product as A. Consider A's entire life cycle. A's expected life cycle structure is represented as a sequence of L consecutive life phases as shown in Figure 8.1.

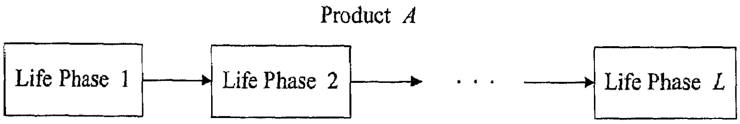


Figure 8.1 Product A's expected life cycle structure.

Life phase k $(k = 1,\cdots,L)$ is represented as a set of n_k timed processes, i.e.,

$$P_k = \{p_1^k(t_{p_1^k} - t_{p_1^k}^0),\cdots,p_i^k(t_{p_i^k} - t_{p_i^k}^0),\cdots,p_{n_k}^k(t_{p_{n_k}^k} - t_{p_{n_k}^k}^0)\},$$

$$t_{p_i^k}^0 \le t_{p_i^k} \le t_{p_i^k}^1,\ i = 1,\cdots,n_k .$$

Selection of P_k is constrained by process sets $P_1,\cdots,\text{and}\,P_{k-1}$ and their corresponding process execution times.

The index function vector of process $p_i^k(t_{p_i^k} - t_{p_i^k}^0)$ is $\mathbf{c}_{p_i^k}(t_{p_i^k} - t_{p_i^k}^0)$. The index vector of life phase k is defined as:

$$\mathbf{c}_{P_k} = \sum_{i=1}^{n_k} \mathbf{c}_{p_i^k}(t_{p_i^k}^1 - t_{p_i^k}^0).$$

The timed IPPD problem is described as follows:

> Find process sets $P_1,P_2,\cdots,P_L$ and their corresponding process execution times $t_{p_i^k}^1 - t_{p_i^k}^0$ $(k = 1,\cdots,L,\ i = 1,\cdots,n_k)$ to optimize criterion $C(\mathbf{c}_{P_1},\mathbf{c}_{P_2},\cdots,\mathbf{c}_{P_L})$.

8.1.3 Timed Solution Methodology

Based on the timed generic framework, we can then solve the timed IPPD problem using the following application procedure:

Step 1: Set up an expected life cycle structure (ELCS) for product A;
Step 2: Identify whether the ELCS is simple or not. If not, simplify it;
Step 3: Identify a set of interesting characteristics as indices;
Step 4: Set up a timed life locus tree for product A;
Step 5: Search in the tree for an optimal life locus.

Steps 1, 2, and 3 are the same as the original IPPD methodology. Steps 4 and 5 are different due to the introduction of time variable. This section focuses on dealing with these two steps, i.e., timed life locus tree setup and optimal search in a timed tree.

(a) Timed Life Locus Tree

In a timed life locus tree, each process set P_k is associated with a time region which indicates the execution times of its component processes. Figure 8.2 shows the structure of a timed life locus tree. In a timed life locus tree, a node on level k is represented as:

$$\{P_1,\cdots,P_k\} \mid \{R_1^{(k)},\cdots,R_k^{(k)}\},$$

where the selection of P_k is constrained by process sets $\{P_1,\cdots,P_{k-1}\}$ and their associated time regions $\{R_1^{(k-1)},\cdots,R_{k-1}^{(k-1)}\}$, i.e.,

$$P_k \mid_{\{P_1,\cdots,P_{k-1}\}|\{R_1^{(k-1)},\cdots,R_{k-1}^{(k-1)}\}}.$$

For any path from root to leaf, we have $R_j^{(k)} \subseteq R_j^{(k-1)}$.

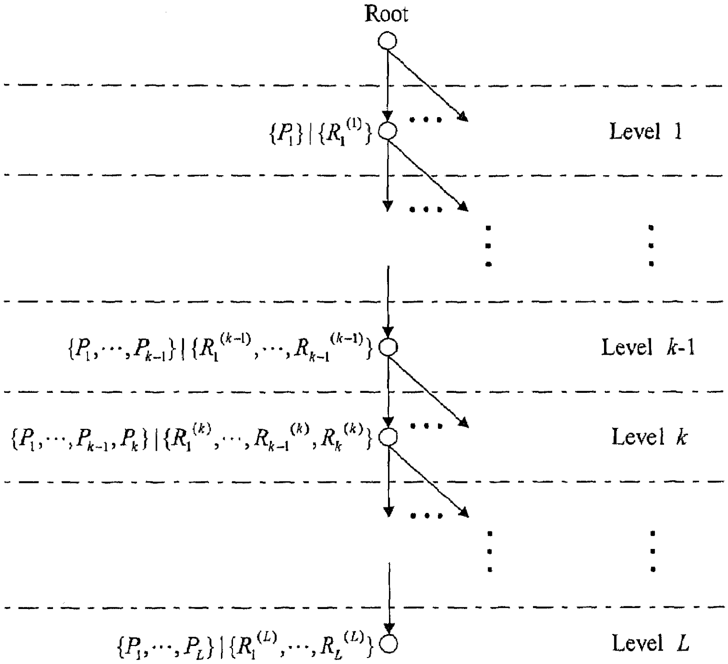


Figure 8.2 A timed life locus tree.

(b) Optimal Search

A timed life locus is represented as:

$$LL = \{P_1,\cdots,P_L\} \mid \{R_1^{(L)},\cdots,R_L^{(L)}\},$$

where $R_k^{(L)}$ is the associated time region of life phase k.

The index vector of life locus LL is defined as:

$$\mathbf{c}_{LL} = \sum_{k=1}^{L} \mathbf{c}_{P_k|R_k^{(L)}},$$

where $\mathbf{c}_{P_k|R_k^{(L)}}$ is defined as:

$$\mathbf{c}_{P_k|R_k^{(L)}} = \underset{R_k^{(L)}}{\text{Optimal}}\{\mathbf{c}_{P_k}\},$$

i.e., the optimal value of $\mathbf{c}_{P_k}$ when its process execution times are in region $R_k^{(L)}$.

Based on this definition of an index vector of a timed life locus, we can then search in the timed life locus tree for an optimal timed life locus. An optimal timed life locus denotes both the selected processes in a product's life cycle and their associated execution times.

8.2 A Case Study on FMS Development

In this section, we apply the timed IPPD methodology to FMS development. For a clear illustration, we follow the application procedure step by step. FMS is the target product to be developed.

Step 1: Set up an ELCS for FMS.

The life cycle for an FMS has similarities to other products. To simplify our discussion, we assume that FMS layout has been determined. Machine selection and decisions along its life are of primary interest. The latter includes, for example, how many times a particular FMS component should be upgraded. Thus, we can use the ELCS of FMS as shown in Figure 8.3.

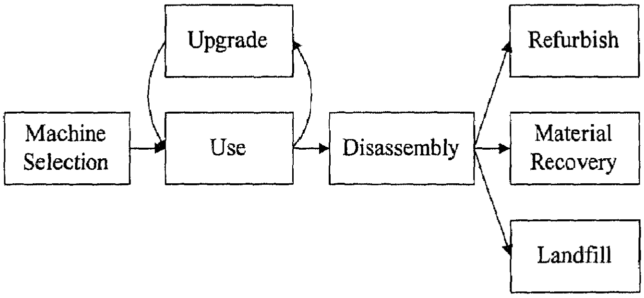


Figure 8.3 An ELCS for FMS.

It begins with its machine selection life phase, followed by its use life phase. When certain components in FMS become technologically obsolescent, upgrades can be carried out in the upgrade life phase. After upgrading, FMS continues its usage. When the entire FMS retires, it enters the disassembly life phase. After disassembly, some components are refurbished for secondary use, some are recovered for material reuse, and the others are landfilled.

Step 2: Identify whether the ELCS is simple or not. If not, simplify it.

Figure 8.4 shows the existence of two complex ELC sub-structures in an FMS's ELCS. Simplifying each of them into a compound life phase, we obtain a simple ELCS as shown in Figure 8.5. It consists of three consecutive life phases: machine selection, use and upgrade, and recovery.

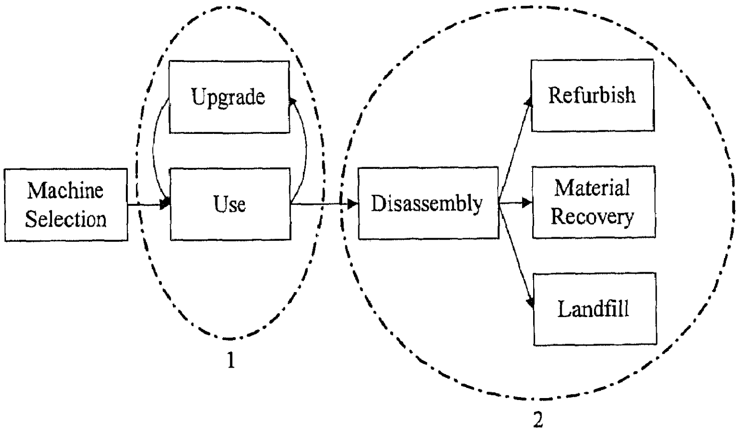


Figure 8.4 Two complex ELC sub-structures in the FMS's ELCS.

Figure 8.5 A simple ELCS for FMS.

Step 3: Identify a set of interesting characteristics as indices.

Selecting an appropriate FMS requires consideration of many factors for successful implementation and operation. Primary factors are the system's flexibility, life cycle cost, and competitive ability over time [Abdel-Malek and Wolf, 1994]. To give proper consideration to these three factors and an FMS's environmental consciousness, we set up an index vector of three indices: cost, benefit, and environmental impact, i.e.,

$$\mathbf{c} = \begin{pmatrix} \text{Cost} \\ \text{Benefit} \\ \text{EI} \end{pmatrix}.$$

The first index characterizes the cost of a certain process; the second index evaluates its performance and competitive ability, e.g., flexibility during usage; and the third index evaluates its associated environmental impact. These three indices varying with time then characterize a process during its execution.

Step 4: Set up a timed life locus tree for FMS.

First we need to identify all the possible processes in each life phase of an FMS's life cycle. Based on the case study of [Abdel-Malek and Wolf, 1994], we have constructed the below FMS development problem and identified the following processes for each life phase.

Machine Selection phase has five different FMS designs corresponding to different FMS component selections. In this case study, we consider six possible components in an FMS, i.e., AGVs (A), robots (R), CNC machines (C), machining centers (M), conveyor systems (V), and database systems (D). Table 8.1 lists the component selections for each FMS design [Abdel-Malek and Wolf, 1994]. For example, FMS design D1 has four components: A, R, C, and M, but does not have V and D.

Suppose that in Use and Upgrade phase, component x needs to be upgraded n_x times. Then there are n_x+1 use processes Ux and n_x upgrade processes UPGx.

Table 8.1 Five FMS designs [Abdel-Malek and Wolf, 1994].

Process	A	R	C	M	V	D
D1	*	*	*	*	-	-
D2	*	*	*	*	-	*
D3	*	*	*	-	-	-
D4	-	*	*	*	*	-
D5	*	*	*	-	-	*

* means selected, - means not selected.

In Recovery phase, each component x (except database) is first disassembled using disassembly process DISx, and then refurbished for secondary use using process REFx with probability ρ_{x1}, or recovered for material reuse using process MRx with probability ρ_{x2}, or landfilled using process LFx with probability ρ_{x3}. If x is upgraded n_x times in Use and Upgrade phase, there will be n_x+1 processes of DISx, REFx, MRx, and LFx in Recovery phase. The values of ρ_{x1}, ρ_{x2}, and ρ_{x3} depend on the execution time of process USEx which determines the status of component x after usage.

Among these possible processes, use processes are regarded as varying execution time processes. All the other processes are regarded as fixed execution time processes.

Then we have to obtain the index function vector for each process. Table 8.2 summarizes all the possible processes in an FMS's life cycle together with their corresponding types and index function vectors. For a fixed execution time process p, index vector $\mathbf{c}_p(t_p^1 - t_p^0)$ is listed instead of its index function vector $\mathbf{c}_p(t_p - t_p^0)$.

Table 8.2 Possible processes in an FMS's life cycle and their types and index (function) vectors.

Life Phase	Process	Type	Cost	Benefit	EI
Machine Selection	D1	Fixed	400	0	0
	D2	Fixed	404	0	0
	D3	Fixed	200	0	0
	D4	Fixed	385	0	0
	D5	Fixed	204	0	0
Use & Upgrade	UA	Varying	$3t$	$B_{UA}(t)$	$0.5t$
	UR	Varying	$3t$	$B_{UR}(t)$	$0.5t$
	UC	Varying	$6t$	$B_{UC}(t)$	t
	UM	Varying	$7t$	$B_{UM}(t)$	t
	UV	Varying	$2t$	$B_{UV}(t)$	$0.5t$
	UD	Varying	t	$B_{UD}(t)$	0
	UPGA	Fixed	24	0	0
	UPGR	Fixed	16	0	0
	UPGC	Fixed	120	0	0
	UPGM	Fixed	160	0	0
	UPGV	Fixed	12	0	0
	UPGD	Fixed	3.2	0	0
Recovery	DISA	Fixed	3	0	1
	DISR	Fixed	3	0	1
	DISC	Fixed	5	0	2
	DISM	Fixed	7	0	3
	DISV	Fixed	2	0	1
	REFA	Fixed	2	10	3
	REFR	Fixed	2	7	3
	REFC	Fixed	5	50	4
	REFM	Fixed	6	67	4
	REFV	Fixed	2	5	3
	MRA	Fixed	3	5	4
	MRR	Fixed	3	3	4
	MRC	Fixed	5	25	7
	MRM	Fixed	5	33	7
	MRV	Fixed	3	2.5	4
	LFA	Fixed	2	0	6
	LFR	Fixed	2	0	6
	LFC	Fixed	5	0	10
	LFM	Fixed	5	0	12
	LFV	Fixed	2	0	6

For varying execution time use process $\mathrm{U}x$, its index functions are as follows:

$$\mathrm{Cost}_{\mathrm{U}x}(t) = \mathrm{C}_{\mathrm{U}x}\, t,$$

$$\mathrm{Benefit}_{\mathrm{U}x}(t) = \sum_{\tau=1}^{t} \left\{ \mathrm{B}_{\mathrm{U}x} / (1+\Phi_x)^{\tau-1} \right\},$$

$$\mathrm{EI}_{\mathrm{U}x}(t) = \mathrm{E}_{\mathrm{U}x}\, t\,,$$

where the discrete time variable t (year) = $1, 2, \cdots$. Its Cost and EI functions are linear. $\mathrm{C}_{\mathrm{U}x}$ and $\mathrm{E}_{\mathrm{U}x}$ are constant values representing its annual operational cost and environmental impact. Its benefit function $\mathrm{Benefit}_{\mathrm{U}x}(t)$ is calculated as the cumulative benefit at year t after considering the technological improvement rate Φ_x of component x. If $\mathrm{B}_{\mathrm{U}x}$ is the initial annual operational benefit, $\mathrm{B}_{\mathrm{U}x}/(1+\Phi_x)^{t-1}$ is the annual operational benefit of year t.

The values of Φ_x for different component x are as follows [Abdel-Malek and Wolf, 1994]:

$$\Phi_{\mathrm{A}} = 0.03,\ \Phi_{\mathrm{R}} = 0.03,\ \Phi_{\mathrm{C}} = 0.025,$$

$$\Phi_{\mathrm{M}} = 0.03,\ \Phi_{\mathrm{V}} = 0.002,\ \Phi_{\mathrm{D}} = 0.10.$$

Finally, a timed life locus tree for FMS is set up as shown in Figure 8.6. The following simplified representations are used:

(1) Only process set P_k is shown on each level of the tree;

(2) On the Recovery level, Rx is used to represent four processes: DISx, REFx, MRx, and LFx;

(3) ρ_{x1}, ρ_{x2}, and ρ_{x3} are determined as follows:

- If the execution time of Ux is less than 10 years, $\rho_{x1} = 0.4$, $\rho_{x2} = 0.5$, and $\rho_{x3} = 0.1$;
- Otherwise, $\rho_{x1} = 0.2$, $\rho_{x2} = 0.4$, and $\rho_{x3} = 0.4$.

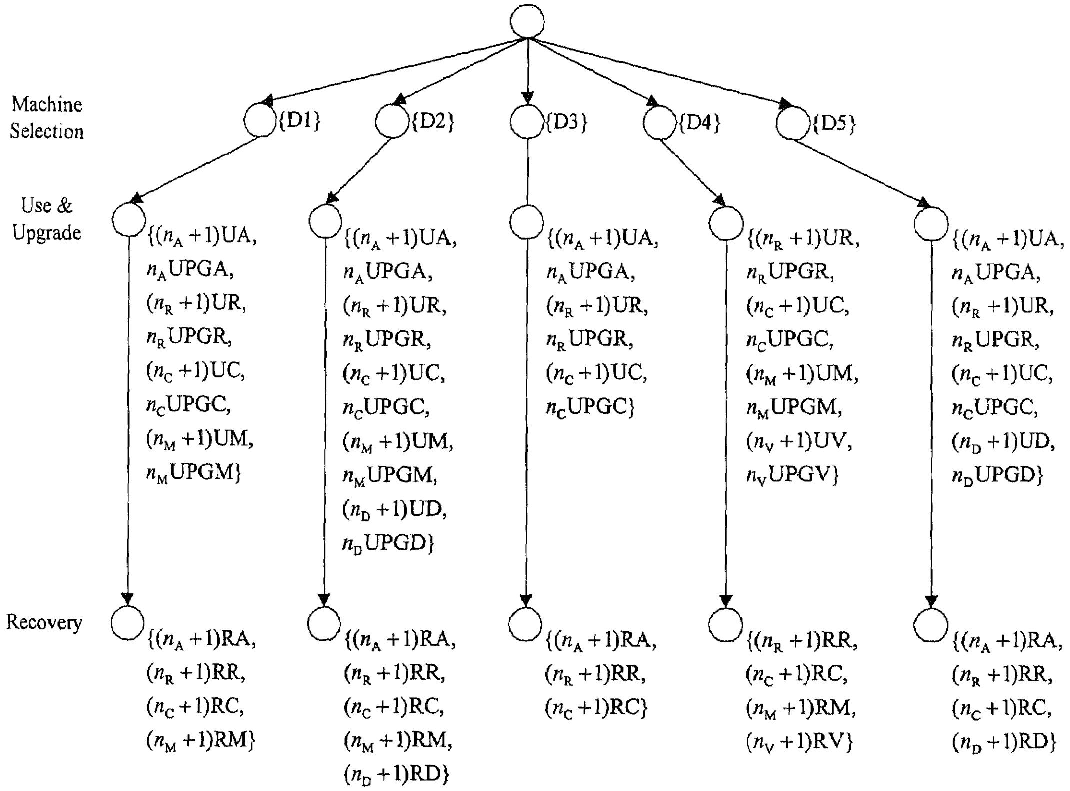


Figure 8.6 A timed life locus tree for FMS.

Step 5: Search in the tree for an optimal life locus.

The optimization criterion is defined as:

$$C = w_1\mathrm{Cost} - w_2\mathrm{Benefit} + w_3\mathrm{EI}.$$

We assume that an FMS is expected to have a life time of 15 years and each FMS component x should not be upgraded for more than 3 times, i.e., $n_x \leq 3$. When n_x is selected, the execution times of the n_x+1 use processes Ux can be determined accordingly, which in turn determine the probabilities of recovery processes REFx, MRx, and LFx. Table 8.3 summarizes the relations among upgrading number n_x, use time t_x, and probabilities ρ_{x1}, ρ_{x2}, and ρ_{x3}.

Table 8.3 Relations among n_x, t_x, ρ_{x1}, ρ_{x2}, and ρ_{x3}.

n_x	t_x	ρ_{x1}	ρ_{x2}	ρ_{x3}
0	15	0.2	0.4	0.4
1	8, 7	0.4	0.5	0.1
2	5, 5, 5	0.4	0.5	0.1
3	4, 4, 4, 3	0.4	0.5	0.1

With these considerations, we propose the following optimal search algorithm on an FMS's timed life locus tree:

(1) For each component x, find its optimal upgrading number n_x^* and use time t_x^* by minimizing $C_x = w_1\text{Cost}_x - w_2\text{Benefit}_x + w_3\text{EI}_x$, where Cost_x, Benefit_x, and EI_x are the life cycle indices of component x;

(2) Based on n_x^* and t_x^*, calculate the optimal criterion value C_i^* for each FMS design Di;

(3) Select an optimal $C^* = \min_{i=1,\cdots,5}\{C_i^*\}$.

When different weighting factors are selected, we obtain different results. Table 8.4 lists some typical search results (i.e., optimal life loci).

Table 8.4 Optimal life loci for an FMS.

w_1, w_2, w_3	1, 1, 1	10, 1, 1	1, 10, 1	1, 1, 10
C	Cost - Benefit + EI	10Cost - Benefit + EI	Cost - 10Benefit + EI	Cost - Benefit + 10EI
Optimal Design	D2	D5	D2	D2
n_A^*	1	0	3	0
n_R^*	1	0	3	0
n_C^*	1	0	3	0
n_M^*	1	-	3	0
n_V^*	-	-	-	-
n_D^*	3	2	3	3
Cost* Benefit* EI*	1108.8 4952.9 119.4	424.4 2533.9 46.4	1824.0 5347.9 193.8	743.8 4432.4 70.4
C^*	-3724.7	1756.5	-51461.2	-2984.6

Table 8.4 (continued)

w_1 , w_2 , w_3	1, 0, 0	0, 1, 0	0, 0, 1
C	Cost	- Benefit	EI
Optimal Design	D3	D2	D3
n_A^*	0	3	0
n_R^*	0	3	0
n_C^*	0	3	0
n_M^*	-	3	-
n_V^*	-	-	-
n_D^*	-	3	-
Cost*	399.0	1824.0	399.0
Benefit*	2033.5	5347.9	2033.5
EI*	46.4	193.8	46.4
C^*	399.0	-5347.9	46.4

8.3 Summary

This chapter introduces a timed IPPD methodology and applies it to the life cycle development of flexible manufacturing systems (FMSs). The timed IPPD methodology considers the execution duration of processes and their time-varying characteristics, and therefore has increased modeling capability and decision accuracy. A detailed case study on FMS development is performed considering its cost, benefit, and environmental impact. FMS machine selection and decisions along its life are considered.

The proposed approach differs from traditional FMS machine selection approaches in that it not only considers FMS machine selection phase, but also considers its use, upgrade, disassembly, and end-of-life recovery phases, i.e., the entire life cycle of an FMS. For example, given an FMS's expected life time, this approach can determine the optimal number of upgrades for each FMS component.

Another characteristic of this approach is that environmental consciousness becomes a consistent part of decision-making in FMS development. It provides a new way to develop cost-effective, high-quality, and environmentally conscious FMSs.

CHAPTER 9

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

9.1 Conclusions

Sustainable development is a strategy by which communities seek economic development approaches that also benefit the local environment and quality of life. It has become an important guide to many communities that have discovered that traditional approaches to planning and development are creating, rather than solving, societal and environmental problems. Where traditional approaches can lead to congestion, sprawl, pollution, and resource overconsumption, sustainable development offers real, lasting solutions that will strengthen our future.

Sustainable development provides a framework under which communities can use resources efficiently, create efficient infrastructures, protect and enhance quality of life, and create new businesses to strengthen their economies. It aims at meeting the needs of the present without compromising the ability of future generations to meet their own needs. Integrated product and process development and design for environment are two important approaches to sustainable development.

Integrated product and process development (IPPD), also called life cycle engineering or concurrent engineering, is a systematic approach to the integrated, concurrent design of products and their related processes. It is intended to cause the developers, from the outset, to consider all the elements of a product's life cycle from conception through disposal, including quality, cost, schedule, and user requirements. Design for environment (DFE) is a systematic consideration of design performance with respect to environmental, health, and safety objectives over the full product and process life cycle. It takes place early in a product's design phase to ensure that the environmental consequences of a product's life cycle are understood before manufacturing decisions are committed. This research focuses on IPPD methodologies for environmentally conscious electronic products.

After a review of current research issues in the field of product and process development, we notice that most of these issues have the same basic nature, i.e., performance optimization while meeting the requirements exerted on a product. Based on this observation, a generic framework for IPPD is proposed. In this framework, a product's entire life cycle is considered and divided into a sequence of life phases. Each life phase transforms the product status from one to another and is represented as a set of processes. A general index vector is applied to each life phase to evaluate its performance. It may include such components as cost, benefit, and environmental impact. The index vectors are then gathered into an index matrix to evaluate the performance of a product's entire life cycle. The index matrix and its various transforms then serve as the optimization objective for different purposes. By this framework, most of the concerned issues can be described formally as constrained optimization problems.

Based on this generic framework, an IPPD methodology is introduced. Important concepts of the methodology are defined and an application procedure is provided illustrating how to apply the methodology systematically to real product and process development. A case study is also performed by applying the methodology to personal computer development. The case study shows that the proposed IPPD methodology can be applied to "green" product and process development meaningfully and efficiently.

In order to have a concrete evaluation of environmental impact of products and processes, eco-compass concept is integrated into the methodology. Eco-compass evaluates environmental impact using six indices: mass intensity, energy intensity, health and environmental potential risk, revalorization, resource conservation, and service extension. Plus cost and benefit, an eight-index vector is set up to evaluate the performance of processes, life phases, and a product's different life loci. As an example, we consider the development of a business telephone. By applying the proposed approach, eco-compass LCA data provided by MERC's LCA research group, and some assumed data due to its unavailability, we can select the optimal telephone design and its associated production, usage, and recovery processes. The case study can serve as a benchmark study example for different IPPD approaches.

To automate the design of competitive and environmentally conscious products and processes, we then propose to describe relations among processes and all related rules formally using logic. A logical deduction algorithm is proposed to generate automatically all the feasible life loci for a target product. Using this approach, product developers need only consider all the logical relations among processes in a product's life cycle and the rest is automatically performed. The efficiency of this approach is demonstrated through two product development examples. This work is critically important to the automation of IPPD processes.

Finally, we extend this research to the life cycle development of flexible manufacturing systems (FMSs). Since implementing an FMS is very costly and the investment tends to be irreversible, careful consideration is required before a decision can be made. To cope with this problem, we introduce a timed IPPD methodology. The execution duration of processes and their time-varying characteristics are considered to increase its modeling capability and decision accuracy. A case study on FMS development is then performed considering its cost, benefit, and environmental impact. FMS machine selection and other decision issues along its life are considered. This approach provides a new way to develop cost-effective, high-quality, and environmentally conscious FMSs.

Application of the proposed IPPD methodologies has the following advantages:

(1) The methodologies provide an integrated decision support mechanism for product developers;

(2) Non-functional requirements for a product are abstracted into formal mathematical representations and are fulfilled using the same mechanism;

(3) Environmental consciousness, among many other considerations, are easily integrated into product development;

(4) Application of the methodologies can reduce product development time and therefore increase the competitiveness of a company in the global market.

This research also has some limitations:

(1) Because of the inherent complexity of product development, efficient optimization techniques are needed when using the methodologies in real applications;

(2) Much professional knowledge and expertise are required to apply the methodologies successfully, and an extensive data collection is needed to obtain the valid data.

9.2 Future Research Directions

Further intensive research is required in many aspects. Some directions are:

(1) Theoretical Research

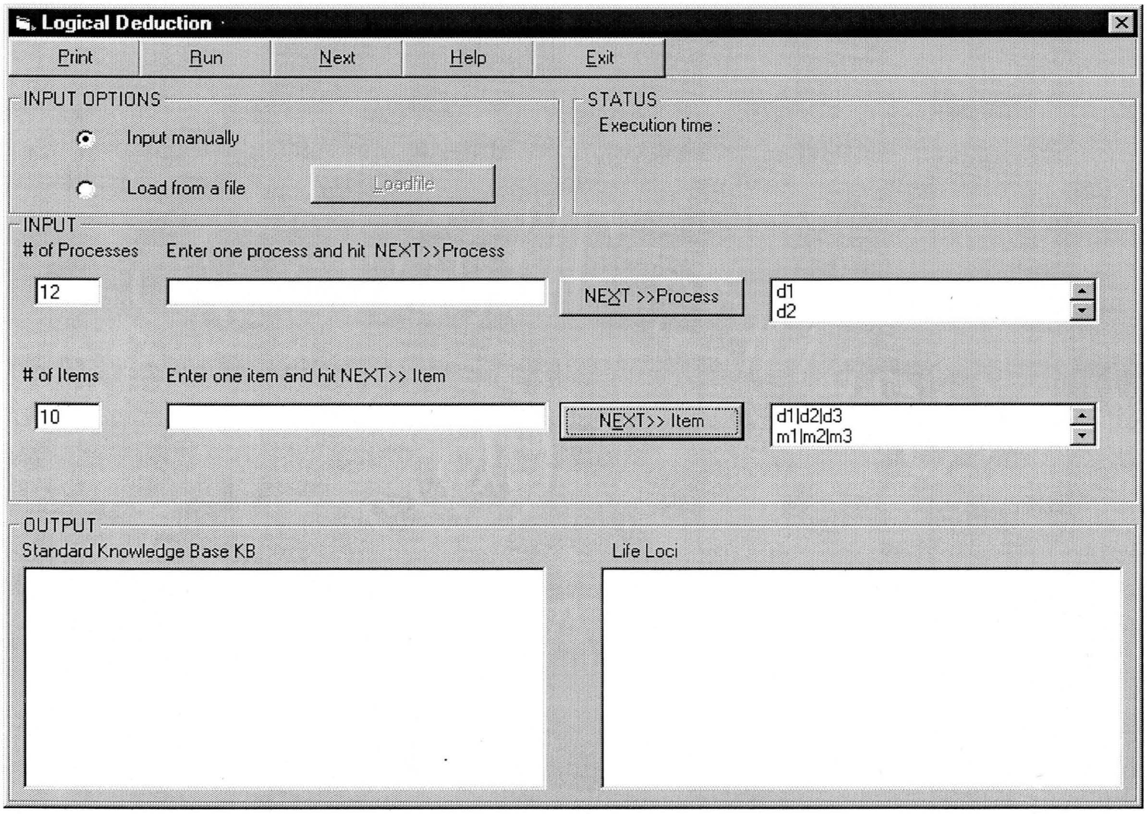
This research mainly focuses on providing system level solutions to efficient and environmentally conscious product and process development. Detailed theoretical research is needed in many specific areas. Some important areas include:

- Hierarchical framework: The proposed methodologies are efficient when solving small to medium-sized IPPD problems. When large problems are encountered, hierarchical framework and solution methods are needed. Petri nets and heuristic approaches are a possible direction;
- Multi-objective optimization: In the proposed methodologies, multiple objectives were transformed into a single objective by the weighting method and the optimization is performed using exhaustive search. In order to use these methodologies successfully in practical applications, efficient multi-objective optimization techniques deserve considerable attention and research effort. Genetic algorithms and neural networks are two possible directions;
- Design synthesis: The proposed methodologies provide an integrated decision support mechanism for product developers. Based on the analysis results, valuable design advice may be obtained, e.g., through sensitivity analysis or as a kind of information feedback, to help synthesize new designs or improve the current designs; and
- Heterogeneous indices: In the proposed methodologies, we assume that each index is evaluated using the same unit (e.g., US dollar). This assumption may not be true

or appropriate when certain indices are more accurately evaluated using fuzzy, stochastic, or incomparable units. Therefore, extension should be incorporated to allow heterogeneous indices in the optimization/solution processes.

(2) Software Development

A software prototype for IPPD based on logical representation of process relations (Chapter 7) has been developed using Visual Basic and Microsoft Access. Using this software, product developers first input a knowledge base *KB* either manually or from a Microsoft Access database. *KB* is then standardized and all the feasible life loci are generated using the logical deduction algorithm. Finally, a depth-first search algorithm is used to search for the optimal life loci. Some interface examples are as shown in Figures 9.1-9.3 [Khatri, Yan, and Zhou, 1999].



● Input manually

○ Load from a file

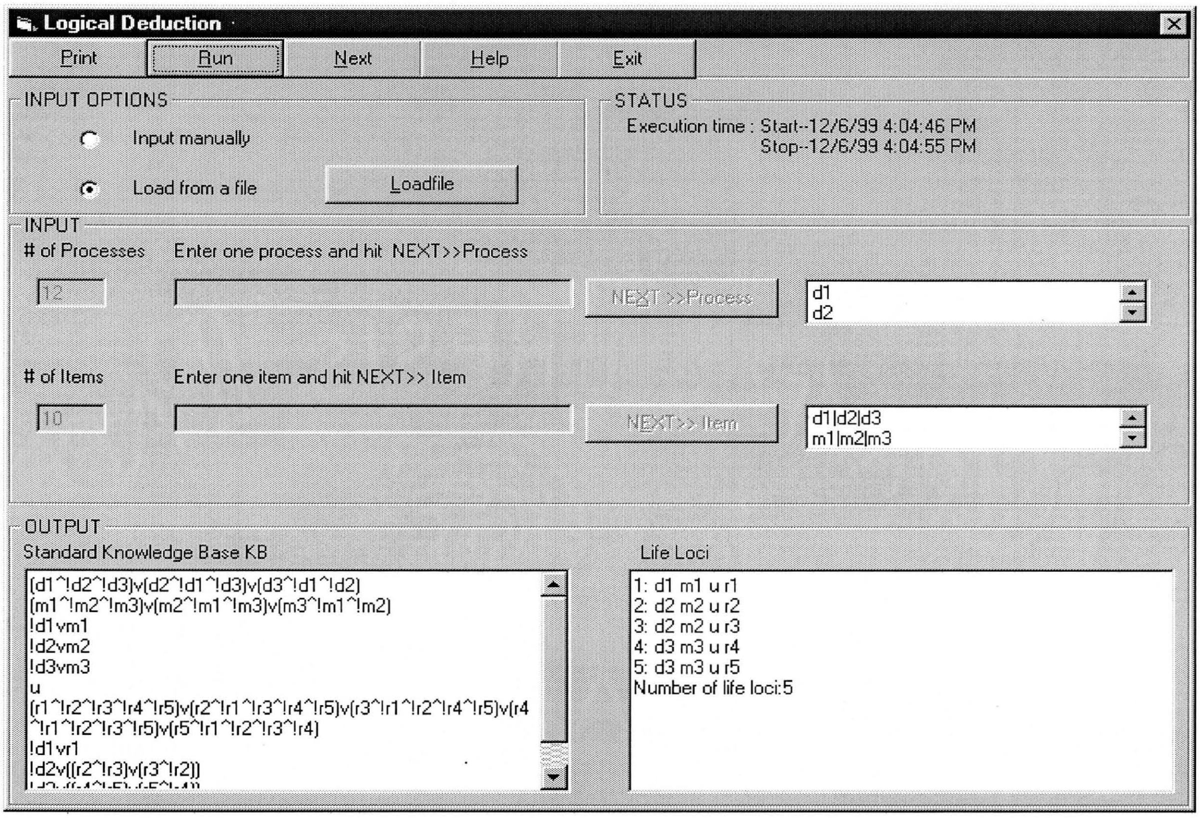
# of Processes 12

# of Items 10

d1
d2

d1|d2|d3
m1|m2|m3

Figure 9.1 Input a knowledge base *KB* .



INPUT OPTIONS

○ Input manually

● Load from a file

STATUS

Execution time : Start--12/6/99 4:04:46 PM
Stop--12/6/99 4:04:55 PM

INPUT

# of Processes: 12

Enter one process and hit NEXT>>Process

```
d1
d2
```

# of Items: 10

Enter one item and hit NEXT>> Item

```
d1|d2|d3
m1|m2|m3
```

OUTPUT

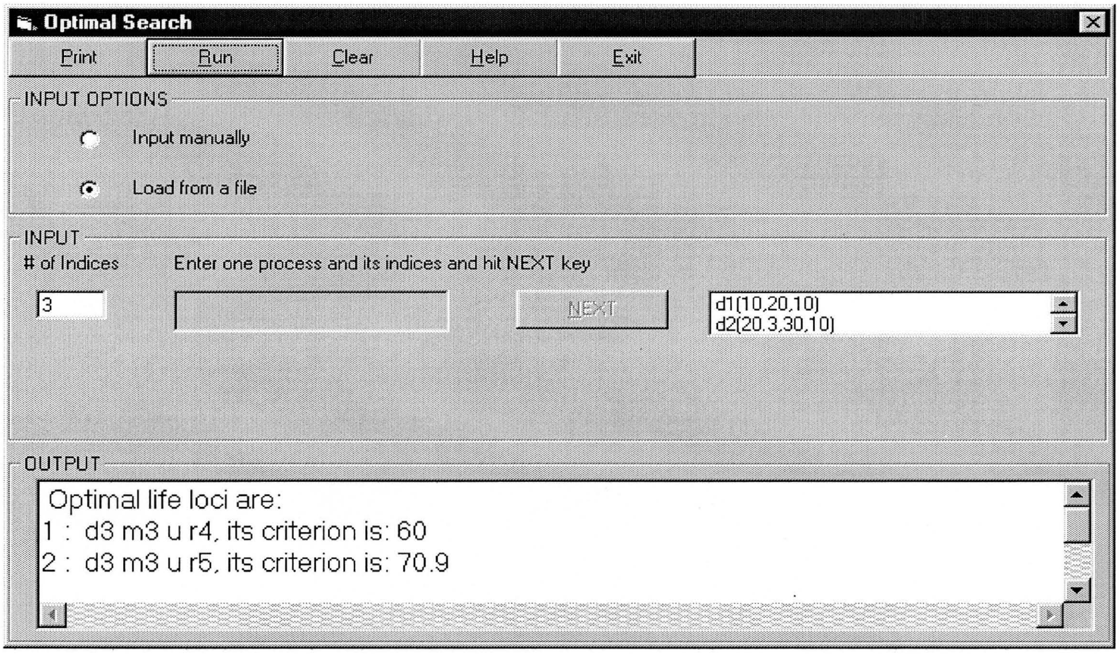
Standard Knowledge Base KB

```
(d1^!d2^!d3)v(d2^!d1^!d3)v(d3^!d1^!d2)
(m1^!m2^!m3)v(m2^!m1^!m3)v(m3^!m1^!m2)
!d1vm1
!d2vm2
!d3vm3
u
(r1^!r2^!r3^!r4^!r5)v(r2^!r1^!r3^!r4^!r5)v(r3^!r1^!r2^!r4^!r5)v(r4
^!r1^!r2^!r3^!r5)v(r5^!r1^!r2^!r3^!r4)
!d1vr1
!d2v((r2^!r3)v(r3^!r2))
```

Life Loci

```
1: d1 m1 u r1
2: d2 m2 u r2
3: d2 m2 u r3
4: d3 m3 u r4
5: d3 m3 u r5
Number of life loci:5
```

Figure 9.2 Standardize *KB* and perform automatic life locus setup.



INPUT OPTIONS

○ Input manually

● Load from a file

INPUT

# of Indices: 3

Enter one process and its indices and hit NEXT key

```
d1(10,20,10)
d2(20.3,30,10)
```

OUTPUT

```
Optimal life loci are:
1 :  d3 m3 u r4, its criterion is: 60
2 :  d3 m3 u r5, its criterion is: 70.9
```

Figure 9.3 Depth-first optimal search.

To facilitate the computer input, we use ! for $\neg$, ^ for $\wedge$, v for $\vee$, > for $\rightarrow$, and | for $\uparrow$.

More effort is needed to further implement the proposed methodologies into a software package. Using today's advanced computing technologies, the software package can provide product developers with a set of systematic approaches to environmentally conscious product and process development and therefore increase significantly the efficiency and profitability of the manufacturing and demanufacturing industries.

(3) Industrial Case Studies

Although several case studies are performed in this research, they are limited to laboratory test beds. Collaboration with industrial firms in some real product development applications would be beneficial to advancing the theoretical research results into practical use.

REFERENCES

[1] L. Abdel-Malek and C. Wolf, "Measuring the impact of lifecycle costs, technological obsolescence, and flexibility in the selection of FMS design," *J. of Manufacturing Systems*, vol. 13, pp. 37-47, 1994.

[2] M. K. Agarwala, A. Bandyopadhyay, and R. van Weeren, "FDC, rapid fabrication of structural components," *American Ceramic Society Bulletin*, vol. 75, pp. 60-65, Nov. 1996.

[3] C. W. Allen, *Simultaneous Engineering: Integrating Manufacturing and Design*. Dearborn, MI: Society of Manufacturing Engineers, 1990.

[4] H. F. Al-Okush, *Assessing the Impact of Design for Environment Guidelines: A Case Study of Office Telephones*. Master's Thesis, Department of Industrial and Manufacturing Engineering, New Jersey Institute of Technology, Newark, NJ, Dec. 1998.

[5] H. Al-Okush, R. J. Caudill, and V. Thomas, "Understanding the real impact of DFE guidelines: A case study of four generations of telephones," in *Proc. IEEE Int. Symp. on Electronics and the Environment*, Danvers, MA, May 1999, pp 134-139.

[6] L. Alting, M. Hauschild, and H. Wenzel, "Elements in a new sustainable industrial culture. Environmental assessment in product development," *Robotics and Computer-Integrated Manufacturing*, vol. 14, pp. 429-439, 1998.

[7] C. H. Amon, J. L. Beuth, and L. E. Weiss, "Shape deposition manufacturing with microcasting: Processing, thermal and mechanical issues," *Journal of Manufacturing Science and Engineering*, vol. 120, pp. 656-665, Aug. 1998.

[8] A. Azapagic and B. Solberg Johansen, "Life cycle assessment," *Engineering* (London, England), vol. 239, pp. 26-27, Feb. 1998.

[9] A. Bandyopadhyay, R. K. Panda, and V. F. Janas, "Processing of piezocomposites by fused deposition technique," *Journal of the American Ceramic Society*, vol. 80, pp. 1366-1372, June 1997.

[10] J. R. Barton, D. Dalley, and V. S. Patel, "Life cycle assessment for waste management," *Waste Management*, vol. 16, pp. 35-50, 1996.

[11] C. Bates, "Tractor production rides on cellular machining system," *American Machinist*, vol. 141, p. 126+, Oct. 1997.

[12] J. Bavishi, "Estimating software as a DFM tool," *Manufacturing Engineering*, vol. 118, p. 92+, Mar. 1997.

[13] K. S. Betts, "Preventing pollution by design," *Environmental Science and Technology*, vol. 32, pp. 318A-320A, July 1998.

[14] L. Beu and L. Mendicino, "Application of design for environment concepts to the semiconductor industry," *Environmental Progress*, vol. 16, pp. 245-250, 1997.

[15] J. R. Blau, "Europe turns competitive with concurrent engineering," *Machine Design*, vol. 66, pp. 41-42+, June 1994.

[16] P. R. Bonenberger, "Stretching the limits of DFM," *Machine Design*, vol. 66, pp. 67-70, Sept. 1994.

[17] A. Brinkley, J. R. Kirby, and F. Charron, "A joint industry approach to environmental impact evaluation of electrical and electronic products," in *Proc. IEEE Int. Symp. on Electronics and the Environment*, San Francisco, CA, May 1997, pp. 309-311.

[18] F. D. Buggie, *New Product Development Strategies*. New York: AMACOM, 1981.

[19] M. Cantamessa, "The manufacturing system as a complex artifact," *Robotics and Computer-Integrated Manufacturing*, vol. 14, pp. 403-414, 1998.

[20] R. J. Caudill *et al.*, "Multi-lifecycle engineering and demanufacturing of discarded electronic products," *Technical Report*, Multi-lifecycle Engineering Research Center, New Jersey Institute of Technology, Newark, NJ, Oct. 1998.

[21] R. J. Caudill *et al.*, *A Final Proposal to the National Science Foundation Engineering Research Centers Program*. Multi-lifecycle Engineering Research Center, New Jersey Institute of Technology, Newark, NJ, Sept. 1997.

[22] M. Chalmers, "Model construction in the fast lane," *Automotive Engineer*, vol. 23, pp. 61-62, Oct. 1998.

[23] K. Chan, C. King, and P. Wright, "COMPASS: Computer Oriented Materials, Processes, and Apparatus Selection System," *Journal of Manufacturing Systems*, vol. 17, pp. 275-286, 1998.

[24] Y. M. Chen and J. J. Liu, "Cost-effective design for injection molding," *Robotics and Computer-Integrated Manufacturing*, vol. 15, pp. 1-21, 1999.

[25] S. T. Chubbs and B. A. Steiner, "Life cycle assessment in the steel industry," *Environmental Progress*, vol. 17, pp. 92-95, 1998.

[26] G. M. Considine, "Design rescues products from the scrap heap," *Design News*, vol. 50[51], pp. 25-26, May 1995.

[27] R. H. Crawford and J. J. Beaman, "Solid freeform fabrication: A new manufacturing paradigm," *IEEE Spectrum*, vol. 36, pp. 34-43, Feb. 1999.

[28] C. T. Culbreth, M. Miller, and P. O'Grady, "A concurrent engineering system to support flexible automation in furniture production," *Robotics and Computer-Integrated Manufacturing*, vol. 12, pp. 81-91, 1996.

[29] M. A. Curran, "Broad-based environmental life cycle assessment," *Environmental Science and Technology*, vol. 27, pp. 430-436, Mar. 1993.

[30] M. A. Curran, "Using LCA-based approaches to evaluate pollution prevention," *Environmental Progress*, vol. 14, pp. 247-253, Nov. 1995.

[31] S. K. Das, "A facility layout method for flexible manufacturing systems," *Int. J. of Production Research*, vol. 31, pp. 279-297, 1993.

[32] D. Deitz, "Design advisors," *Mechanical Engineering*, vol. 120, p. 16+, July 1998.

[33] P. De Langhe, S. Criel, and D. Ceuterick, "Green design of telecom products: The ADSL high speed modem as a case study," *IEEE Transactions on Components, Packaging and Manufacturing Technology. Part A, Manufacturing Technology*, vol. 21, pp. 154-167, Mar. 1998.

[34] R. de Graaf and L. Kornelius, "Inter-organizational concurrent engineering: A case study in PCB manufacturing," *Computers in Industry*, vol. 30, pp. 37-47, Sept. 1996.

[35] R. Dominach, "Developing an efficient DFM strategy," *Computer Design*, vol. 32, pp. 77-78+, Oct. 1993.

[36] S. Dowlatshahi, "The role of logistics in concurrent engineering," *International Journal of Production Economics*, vol. 44, pp. 189-199, July 1996.

[37] V. G. Duffy and G. Salvendy, "The impact of organizational ergonomics on work effectiveness: With special reference to concurrent engineering in manufacturing industries," *Ergonomics*, vol. 42, pp. 614-637, Apr. 1999.

[38] P. Dvorak, "Manufacturing puts a new spin on design," *Machine Design*, vol. 66, p. 67+, Aug. 1994.

[39] M. I. Elboushi and J. S. Sherif, "Object-oriented software design utilizing Quality Function Deployment," *Journal of Systems and Software*, vol. 38, pp. 133-143, Aug. 1997.

[40] N. N. Eldin, "Concurrent engineering: A schedule reduction tool," *Journal of Construction Engineering and Management*, vol. 123, pp. 354-362, Sept. 1997.

[41] H. A. ElMaraghy and T. Ravi, "Modern tools for the design, modelling and evaluation of flexible manufacturing systems," *Robotics and Computer-Integrated Manufacturing*, vol. 9, pp. 335-340, 1992.

[42] I. Erikkson and F. McFadden, "Quality function deployment: A tool to improve software quality," *Information and Software Technology*, vol. 35, pp. 491-498, Sept. 1993.

[43] R. Farrell, "Quality function deployment: Helping business identify and integrate the voice of the customer," *Industrial Engineering*, vol. 26, pp. 44-45, Oct. 1994.

[44] G. Ferrer, "The economics of personal computer remanufacturing," *Resources, Conservation and Recycling*, vol. 21, pp. 79-108, 1997.

[45] J. Fiksel, *Design for Environment: Creating Eco-Efficient Products and Processes*. New York: McGraw-Hill, 1996.

[46] C. Fussler and P. James, *Driving Eco Innovation: A Breakthrough Discipline for Innovation and Sustainability*. London: Pitman Publishing, 1996.

[47] R. Gadh and F. B. Prinz, "A computationally efficient approach to feature abstraction in design-manufacturing integration," *Journal of Engineering for Industry*, vol. 117, pp. 16-27, Feb. 1995.

[48] T. F. Gautschi, "Using QFD to develop your product," *Design News*, vol. 48[49], p. 168, Nov. 1993.

[49] W. J. Glantschnig, "Green design: An introduction to issues and challenges," *IEEE Trans. on Components, Packaging, and Manufacturing Technology - Part A*, vol. 17, pp. 508-513, Dec. 1994.

[50] E. A. Glushkovsky, R. A. Florescu, and A. Hershkovits, "Avoid a flop: Use QFD with questionnaires," *Quality Progress*, vol. 28, pp. 57-62, June 1995.

[51] B. L. Goldense, "QFD: Applying 'the 80-20 rule'," *Design News*, vol. 48[49], p. 150, Dec. 1993.

[52] T. E. Graedel and B. R. Allenby, *Industrial Ecology*. New Jersey: Prentice Hall, 1995.

[53] T. E. Graedel, B. R. Allenby, and P. R. Comrie, "Matrix approaches to abridged life cycle assessment," *Environmental Science and Technology*, vol. 29, pp. 134A-139A, Mar. 1995.

[54] B. Graessel and P. Zeidler, "Using quality function deployment to improve customer service," *Quality Progress*, vol. 26, pp. 59-63, Nov. 1993.

[55] N. Green and M. Reder, "DFM boosts U.S. car quality," *Machine Design*, vol. 65, pp. 61-64, May 1993.

[56] R. M. Hainer, S. Kingsbury, and D. B. Gleicher, *Uncertainty in Research, Management, and New Product Development.* New York: Reinhold Pub. Corp., 1967.

[57] R. Hales, "QFD: A key enabling technology in today's advanced product development environments," *Industrial Engineering*, vol. 26, pp. 10-11, Dec. 1994.

[58] M. Harsch, M. Schuckert, and P. Eyerer, "Life-cycle assessment," *Advanced Materials and Processes*, vol. 149, pp. 43-46, June 1996.

[59] M. Hatch and R. D. Badinelli, "A concurrent optimization methodology for concurrent engineering," *IEEE Transactions on Engineering Management*, vol. 46, pp. 72-86, Feb. 1999.

[60] C. L. Havener, "Improving the quality of quality," *Quality Progress*, vol. 26, pp. 41-44, Nov. 1993.

[61] N. J. Hazel, "Using LCA to select coatings for optimum environmental and cost performance," *Journal of Coatings Technology*, vol. 68, pp. 117-121, Oct. 1996.

[62] C. Hendrickson, A. Horvath, and S. Joshi, "Economic input-output models for environmental life-cycle assessment," *Environmental Science and Technology*, vol. 32, pp. 184A-191A, Apr. 1998.

[63] S. S. Heragu and A. Kusiak, "Machine layout problem in flexible manufacturing systems," *Operations Research*, vol. 36, pp. 258-268, 1988.

[64] M. A. Hersh, "A survey of systems approaches to green design with illustrations from the computer industry," *IEEE Trans. on Sys., Man, and Cyber.*, vol. 28, pp. 528-540, Nov. 1998.

[65] G. Q. Huang and K. L. Mak, "Re-engineering the product development process with 'design for X'," in *Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture,* vol. 212, pp. 259-268, 1998.

[66] V. D. Hunt, *Reengineering: Leveraging the Power of Integrated Product Development.* Essex Junction, VT: Omneo, 1993.

[67] K. Ishii, "Life-cycle engineering for design," in *Proc. Quality Concepts Conf.*, Engineering Society of Detroit, MI, Oct. 1990, pp. 194-210.

[68] K. Ishii, C. F. Eubanks, and P. Di Marco, "Design for product retirement and material life-cycle," *Materials and Design*, vol. 15, pp. 225-233, 1994.

[69] J. Jin, *Multi-lifecycle Assessment Design Tools and Software Development.* Master's Thesis, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, May1999.

[70] F. Jovane, L. Alting, A. Armillotta, W. Eversheim, K. Feldmann, G. Seliger, and N. Roth, "A key issue in product life cycle: Disassembly," *Annals of the CIRP*, vol. 42, pp. 651-658, 1993.

[71] U. Kalyan-Seshu and B. Bras, "Towards computer aided design for the life cycle," in *Proc. IEEE Int. Symp. on Electronics and the Environment,* Oak Brook, IL, May 1998, pp. 310-315.

[72] G. A. Keoleian and D. Menerey, "Sustainable development by design: Review of life cycle design and related approaches," *Air and Waste*, vol. 44, pp. 645-668, May 1994.

[73] A. Khatri, P. Yan, and M. Zhou, "A software prototype for IPPD based on logical representation of process relations," *Technical Report*, Multi-lifecycle Engineering Research Center, New Jersey Institute of Technology, Newark, NJ, Dec. 1999.

[74] M. G. King, "Total vehicle development with DFM," *Mechanical Engineering*, vol. 118, pp. 72-74, May 1996.

[75] I. H. Knoepfel, "A framework for environmental impact assessment of long-distance energy transport systems," *Energy* (Oxford, England), vol. 21, pp. 693-702, 1996.

[76] G. Kobe, "Beyond DFA," *Automotive Industries*, vol. 174, p. 64, May 1994.

[77] J. P. Langan, "Design for manufacturability," *Plating and Surface Finishing*, vol. 85, pp. 44-45, Oct. 1998.

[78] R. Langdon, "Rapid prototyping," *Automotive Engineer*, vol. 24, pp. 44-46+, Mar. 1999.

[79] L. B. Lave, E. Cobas Flores, and C. T. Hendrickson, "Using input-output analysis to estimate economy-wide discharges," *Environmental Science and Technology*, vol. 29, pp. 420A-426A, Sept. 1995.

[80] M. Litsikas, "Break old boundaries with concurrent engineering," *Quality*, vol. 36, pp. 54-56, Apr. 1997.

[81] J. C. Lu, W. C. Holton, and J. S. Fenner, "A new device design methodology for manufacturability," *IEEE Transactions on Electron Devices*, vol. 45, pp. 634-642, Mar. 1998.

[82] Y. Luo, Z. Ji, M. C. Leu, and R. Caudill, "Environmental performance analysis of solid freeform fabrication processes," in *Proc. IEEE Int. Symp. on Electronics and the Environment*, Danvers, MA, May 1999, pp. 1-6.

[83] P. Mannion, "IC demands force concurrent engineering," *Electronic Design*, vol. 46, pp. 129-130+, June 1998.

[84] A. Manthiram, D. L. Bourell, and H. L. Marcus, "Nanophase materials in solid freeform fabrication," *JOM*, vol. 45, pp. 66-70, Nov. 1993.

[85] H. L. Marcus and D. L. Bourell, "Solid freeform fabrication finds new applications," *Advanced Materials and Processes*, vol. 144, pp. 28-31+, Sept. 1993.

[86] R. D. Meller and K. Y. Gau, "The facility layout problem: Recent and emerging trends and perspectives," *Journal of Manufacturing Systems*, vol. 15, pp. 351-366, 1996.

[87] C. Mizuki *et al.*, "Electronic products recycling study: Executive summary," *MCC report*, http://www.mcc.com/projects/env/EPRexec.html, 1997.

[88] T. Mueller, "When it comes to new product development, rapid prototyping = risk reduction," *Molding Systems*, vol. 57, pp. 40-46, Apr. 1999.

[89] A. A. Munoz and P. Sheng, "An analytical approach for determining the environmental impact of machining processes," *Journal of Materials Processing Technology*, vol. 53, pp. 736-758, Sept. 1995.

[90] A. S. Munro, "Keys to DFA success," *Machine Design*, vol. 67, p. 162, July 1995.

[91] C. J. Murray, "Design for manufacturing," *Design News*, vol. 50[51], pp. 62-63, Aug. 1995.

[92] S. Myint and M. T. Tabucanon, "A multiple-criteria approach to machine selection for flexible manufacturing systems," *Int. J. of Production Economics*, vol. 33, pp. 121-131, 1994.

[93] R. N. Natarajan, R. E. Martz, and K. Kurosaka, "Applying QFD to internal service system design," *Quality Progress*, vol. 32, pp. 65-70, Feb. 1999.

[94] D. Navin-Chandra, "ReStar: A design tool for environmental recovery analysis," in *Proc. Int. Conf. on Engineering Design*, Aug. 1993.

[95] S. K. Ong and A. Y. C. Nee, "A systematic approach for analyzing the fixturability of parts for machining," *Journal of Manufacturing Science and Engineering*, vol. 120, pp. 401-408, May 1998.

[96] S. O. Onuh and K. K. B. Hon, "Application of the Taguchi method and new hatch styles for quality improvement in stereolithography," in *Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture*, vol. 212, pp. 461-472, 1998.

[97] T. Ormond, "Rapid prototyping systems shorten design cycles," *EDN*, vol. 38, pp. 114-116+, July 1993.

[98] D. T. Pham and S. S. Dimov, "An approach to concurrent engineering," in *Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture*, vol. 212, pp. 13-27, 1998.

[99] M. L. Philpott, C. S. Warrington, and E. A. Branstad, "A parametric contract modeler for DFM analysis," *Journal of Manufacturing Systems*, vol. 15, pp. 256-267, 1996.

[100] J. Poolton and I. Barclay, "Concurrent engineering assessment: A proposed framework," in *Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture*, vol. 210, pp. 321-328, 1996.

[101] B. Prasad, "Review of QFD and related deployment techniques," *Journal of Manufacturing Systems*, vol. 17, pp. 221-234, 1998.

[102] M. Rajasekharan, B. A. Peters, and T. Yang, "A genetic algorithm for facility layout design in flexible manufacturing systems," *Int. J. of Production Research*, vol. 36, pp. 95-110, 1998.

[103] M. K. Raymond, "Software tackles goliath manufacturing," *American Machinist*, vol. 139, pp. 33-35, Nov. 1995.

[104] A. P. Sage, "Systems engineering and management for industrial ecology and sustainable development," in *Proc. IEEE Int. Conf. Sys., Man, and Cyber.*, Orlando, FL, Oct. 1997, pp. 784-790.

[105] K. Saur, J. Hesselbach, and P. Eyerer, "New directions in life cycle assessment," *Automotive Engineering*, vol. 105, pp. 79-82, June 1997.

[106] M. Schuckert, K. Saur, and H. Florin, "Life cycle analysis: Getting the total picture on vehicle engineering alternatives," *Automotive Engineering*, vol. 104, pp. 49-52, Mar. 1996.

[107] P. Sheng and E. Hertwich, "Indices for comparative waste assessment in environmentally-conscious manufacturing," *Journal of Manufacturing Science and Engineering*, vol. 120, pp. 129-140, Feb. 1998.

[108] S. G. Shina, *Successful Implementation of Concurrent Engineering Products and Processes*. New York: Van Nostrand Reinhold, 1993.

[109] R. Socolow, C. Andrews, F. Berkhout, and V. Thomas, *Industrial Ecology and Global Change.* United Kingdom: Cambridge University Press, 1994.

[110] S. Soyucayli and K. N. Otto, "Simultaneous engineering of quality through integrated modeling," *Journal of Mechanical Design*, vol. 120, pp. 210-220, June 1998.

[111] M. Srinivasan, P. Sheng, and T. Wu, "Development of a scoring index for the evaluation of environmental factors in machining processes: Part I - Health hazard score formulation," *Trans. of NAMRI*, vol. 23, pp. 115-122, 1995a.

[112] M. Srinivasan, P. Sheng, and T. Wu, "Development of a scoring index for the evaluation of environmental factors in machining processes: Part II - Applications," *Trans. of NAMRI*, vol. 23, pp. 123-130, 1995b.

[113] V. Sriraman, G. Winek, and R. Habingreither, "Tooling applications of rapid prototyping," *Journal of Engineering Technology*, vol. 16, pp. 34-38, 1999.

[114] N. L. C. Steele and D. T. Allen, "An abridged life-cycle assessment of electric vehicle batteries," *Environmental Science and Technology*, vol. 32, pp. 40A-46A, Jan. 1998.

[115] J. A. Stuart, "Materials selection for life cycle design," in *Proc. IEEE Int. Symp. on Electronics and the Environment*, Oak Brook, IL, May 1998, pp. 151-158.

[116] J. L. Sullivan, M. M. Costic, and W. Han, "Modifying automotive life-cycle assessment," *Automotive Engineering International*, vol. 106, pp. 69-71, July 1998.

[117] C. S. Syan and U. Menon, *Concurrent Engineering: Concepts, Implementation and Practice.* New York: Chapman & Hall, 1994.

[118] J. Talavage and R. G. Hannam, *Flexible Manufacturing Systems in Practice: Applications, Design, and Simulation.* New York: M. Dekker, 1988.

[119] M. V. Tatikonda, "Design for assembly: A critical methodology for product reengineering and new product development," *Production and Inventory Management Journal*, vol. 35, pp. 31-37, 1994.

[120] M. Thomson, C. P. Koshland, and D. Lucas, "A health-based decision-making tool for waste destruction," *Environmental Engineering Science*, vol. 14, pp. 163-173, 1997.

[121] T. Tom, "Entire supply chains must standardize electronic data protocols," *Machine Design*, vol. 69, pp. 204, Jan. 1997.

[122] K. T. Ulrich and S. D. Eppinger, *Product Design and Development.* New York: McGraw-Hill, 1995.

[123] W. Wingo, "Government plans to bolster 'green' design," *Design News*, vol. 50[51], p. 39, June 1995.

[124] P. Yan, M. Zhou, and D. Sebastian, "A generic framework for integrated product and process development," *Int. J. of Environmentally Conscious Design and Manufacturing*, vol. 7, no. 3, pp. 47-57, 1998.

[125] P. Yan, M. Zhou, and D. Sebastian, "An integrated product and process development methodology: Concept formulation," *Robotics and Computer-Integrated Manufacturing*, vol. 15, pp. 201-210, 1999.

[126] P. Yan, M. Zhou, D. Sebastian, and R. Caudill, "Integrating eco-compass concept into integrated product and process development," submitted to *Int. J. of Environmentally Conscious Design and Manufacturing* for review, 1999.

[127] P. Yan and M. Zhou, "A solution methodology for integrated product and process development based on logical representation of process relations," submitted to *IEEE Trans. on Sys., Man, and Cyber.* for review, 1999.

[128] P. Yan, M. Zhou, and D. Sebastian, "Formulation of a generic framework for integrated product and process development," in *Proc. IEEE Int. Conf. on Sys., Man, and Cyber.*, San Diego, CA, Oct. 1998, pp. 2609-2614.

[129] P. Yan, M. Zhou, and D. Sebastian, "An integrated product and process development methodology: A case study," in *CD-ROM Rensselaer's Int. Conf. on Agile, Intelligent, and Computer-Integrated Manufacturing*, Troy, NY, Oct. 1998, Paper No. 46.

[130] P. Yan, M. Zhou, and D. Sebastian, "Multi-lifecycle product and process development: Selection of optimal production, usage, and recovery processes," in *Proc. IEEE Int. Symp. on Electronics and the Environment*, Danvers, MA, May 1999, pp. 274-279.

[131] P. Yan, M. Zhou, and R. Caudill, "A life cycle engineering approach to FMS development," submitted to *IEEE Int. Conf. on Robotics and Automation*, Sept. 1999.

[132] A. A. Yassine, K. R. Chelst, and D. R. Falkenburg, "A decision analytical framework for evaluating concurrent engineering," *IEEE Transactions on Engineering Management*, vol. 46, pp. 144-157, May 1999.

[133] B. N. Yetukuri, N. V. Yetukuri, and G. W. Fischer, "SPAW: A design tool for planning a manufacturing process in a concurrent engineering environment," *Computers in Industry*, vol. 32, pp. 79-93, Dec. 1996.

[134] M. Yoshimura, "Concurrent product design and manufacturing," *Control and Dynamic Systems*, vol. 62, pp. 89-128, 1994.

[135] H. C. Zhang, T. C. Kuo, and H. Lu, "Environmentally conscious design and manufacturing: A state-of-the-art survey," *Journal of Manufacturing Systems*, vol. 16, pp. 352-371, 1997.

[136] H. C. Zhang and S. Y. Yu, "An environmentally conscious evaluation/design support tool for personal computers," in *Proc. IEEE Int. Symp. on Electronics and the Environment*, San Francisco, CA, May 1997, pp. 131-136.

[137] M. Zhou, R. J. Caudill, and X. He, "Evaluation of environmentally conscious product designs," in *Proc. IEEE Int. Conf. on Sys., Man, and Cyber.*, San Diego, CA, Oct. 1998, pp. 4057-4062.

[138] M. Zhou, B. Zhang, R. Caudill, and D. Sebastian, "A cost model for multi-lifecycle engineering design," in *Proc. IEEE Int. Symp. on Emerging Technologies and Factory Automation*, Hawaii, Nov. 1996, pp. 385-391.

[139] E. Zussman and M. Zhou, "A methodology for modeling and adaptive planning of disassembly processes," *IEEE Trans. on Robotics and Automation*, vol. 15, pp. 190-194, Feb. 1999.