RAPID ONE-OF-A-KIND PRODUCT DEVELOPMENT: STRATEGIES, ALGORITHMS AND TECHNOLOGY

S. Q. XIE
Rapid One-Of-a-Kind Product Development: Strategies, Algorithms and Technology

By

Shane (Sheng Quan) XIE

In Partial Fulfilment of the Requirements for the Degree of

Doctor of Philosophy in Mechanical Engineering

University Of Canterbury

August 2001

Copyright © 2001 by Sheng quan XIE
Rapid One-Of-a-Kind Product Development: Strategies, Algorithms and Technology
ACKNOWLEDGMENTS

When I started to write this thesis, I still clearly remembered how hard it was when I did my first PhD degree on Mechatronics in Huazhong University of Science and Technology, P.R.China. Although I had the experience of getting a PhD from a leading engineering university in China, getting another PhD degree in a different research area in a different education system was still a great challenge to me. Without the help of my advisors, colleagues and friends, my research work could not have come to this stage. I would like to take this chance to thank them for their help!

First my appreciation is expressed to my advisors Dr. Paul Tu, Dr. David Aitcherson and Associate Professor Richard Dunlop for their continual encouragement, guidance, and patience throughout my graduate study. They have provided generous support and have been the primary motivation for this research. Their insights have guided me to a level of understanding higher than I thought possible. In particular, I owe special thanks to my main supervisor Dr Paul Tu, who opened the world of One-of-a-kind Production and its usage in numerous manufacturing areas for me. His continued guidance and inspiration have also helped me grow both intellectually and personally.

I would like to thank my colleagues in the Department of Mechanical Engineering for their generous help in this research: Professor Harry for proofreading Chapter 3 and Chapter 4 of the thesis; Andrew Shaw for giving me information about PDM (product data management) systems and helping me with my English study; Juinn Jet Kam for helping me to build up the simulation platform for optimal sheet metal cutting and scheduling; Larry J Kirwan and Kian Wee Chew, for helping me to gather product information of injection moulds and to build up a knowledge base. I would like to acknowledge the great support from several manufacturing companies in Christchurch.

The financial support from the Foundation for Research, Science and Technology (FRST) of New Zealand (UOC805) is greatly appreciated. It makes this work possible.

Three years of both research and graduate study was a long way to go. I consider myself blessed to know my wife, Mrs. Hong Huang. Her love and support made these years’ work and life really enjoyable. I offer deepest thanks to her for her understanding and sacrifice throughout this process. I am also in debt to my parents and parents-in-law for what they have done for me. Without their continual support, I would not have got to this point. This dissertation stands as a testament to their support. I am also grateful to Wang Jia Cheng, Liu Aizhen, Fang Li Xing, Chen Qi, Zhang Jianjing for their friendship and kindness and supporting and helping me during the past three years.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>III</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>IV</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>IX</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>X</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>XIII</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>XIV</td>
</tr>
<tr>
<td><strong>CHAPTER 1</strong></td>
<td></td>
</tr>
<tr>
<td>MOTIVATION AND BACKGROUND</td>
<td></td>
</tr>
<tr>
<td>1.1 Rapid One-Of-a-Kind Product Development</td>
<td>1</td>
</tr>
<tr>
<td>1.1.1 The Performance of RPD</td>
<td>2</td>
</tr>
<tr>
<td>1.1.2 Evaluating The PD Process</td>
<td>3</td>
</tr>
<tr>
<td>1.1.3 Benefits of RPD</td>
<td>4</td>
</tr>
<tr>
<td>1.1.4 Drivers for RPD</td>
<td>4</td>
</tr>
<tr>
<td>1.1.5 Emerging Research Issues</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Motivation</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Thesis objectives</td>
<td>7</td>
</tr>
<tr>
<td>1.4 Overview of the Thesis</td>
<td>9</td>
</tr>
<tr>
<td><strong>CHAPTER 2</strong></td>
<td></td>
</tr>
<tr>
<td>THE STATE-OF-THE-ART LITERATURE REVIEW AND A PROPOSED REFERENCE SYSTEM</td>
<td></td>
</tr>
<tr>
<td>STRUCTURE</td>
<td></td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>13</td>
</tr>
<tr>
<td>2.2 Overview of OKP and Internet based Product Development Systems</td>
<td>15</td>
</tr>
<tr>
<td>2.2.1 Review of OKP</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2 Requirements of Next Generation OKP Systems</td>
<td>17</td>
</tr>
<tr>
<td>2.2.3 Role of IRPD</td>
<td>18</td>
</tr>
<tr>
<td>2.3 Review of Internet based PD approaches</td>
<td>19</td>
</tr>
<tr>
<td>2.3.1 Collaborative Product Design</td>
<td>19</td>
</tr>
<tr>
<td>2.3.2 DFX via the Internet</td>
<td>20</td>
</tr>
<tr>
<td>2.3.3 Internet-based Decision Support</td>
<td>21</td>
</tr>
<tr>
<td>2.3.4 Internet based Scheduling, Production Planning and Manufacturing</td>
<td>22</td>
</tr>
<tr>
<td>2.3.5 Internet based CE</td>
<td>23</td>
</tr>
</tbody>
</table>
2.4 Review of IRPD Approaches 24
2.5 Implementing Technologies for Developing IRPD Systems 26
2.6 Research issues and Future Trends facing IRPD in the 21st century 30
2.6.1 Implementing Issues 30
2.6.2 Systems Integration Methodologies 31
2.6.3 CSCW and Interfacing Techniques 32
2.6.4 Global Cost/Lead time Optimisation 33
2.6.5 Virtual Reality for Manufacturing 34
2.7 A Proposed IRPD System 34
2.8 Chapter Conclusions 36

CHAPTER 3
AN INTERNET BASED INTEGRATED PRODUCT DEVELOPMENT SYSTEM FOR RAPID SHEET METAL PART/PRODUCT DEVELOPMENT
3.1 Introduction 38
3.2 The Integrated Product Development System for Sheet Metal Design and Manufacturing 39
3.3 The Internet based Integrated Data Environment 42
3.4 QFD based Global Customer User Interfaces 42
3.5 Simulation Platform 44
3.6 Knowledge bases to Support Intelligent Design and Manufacturing 44
3.7 Case Studies 46
3.7.1 Programming Tools 46
3.7.2 Application Modules Test 46
3.8 Summary of the Chapter 48

CHAPTER 4
AN INFORMATION FRAMEWORK FOR RAPID SHEET METAL PART/PRODUCT DEVELOPMENT
4.1 Backgrounds and Motivation 49
4.2 The Role of the Information Framework 50
4.3 The Zero thickness and Zero Bend Radius Principle 51
4.4 The Information Integration framework 52
4.4.1. The Step Structure Information Framework 52
4.4.2. The Information Relationship Representation of Sheet Metal Parts
4.4.3. Modeling Languages and Methodology
4.5 Case Study
4.6 Chapter Conclusion

CHAPTER 5

COMPOUND SHEET METAL CUTTING METHOD TO SPEED UP
THE SHEET METAL PART/PRODUCT DEVELOPMENT

5.1 Introduction
5.2 Integrated System Architecture for Punching and Cutting
5.3 Data Integration Platform for Sheet Metal Compound Manufacturing
5.4 Knowledge based RTCAPP system for CNC punching and cutting
5.4.1 Tools Sequencing Real Time Optimization Algorithm for Punching
5.4.1.1 Mathematical Model
5.4.1.2 Mathematical Description of The Shortest Insert Algorithm
5.4.1.3 Programming Implementation
5.4.1.4 The Performance of the algorithm
5.4.2 Knowledge Based CAPP for Sheet Metal Cutting
5.4.2.1 “Cut-in” and “cut-out” auxiliary path
5.4.2.2 least distance between “start hole” and contour
5.4.3 Tool Path Optimisation for Cutting and Cost Estimation
5.5 Simulation Platform for Compound Manufacturing
5.6 Conclusions of the Chapter

CHAPTER 6

AN INTERNET BASED COST ESTIMATION AND OPTIMISATION
FRAMEWORK FOR SHEET METAL PART/PRODUCT
DEVELOPMENT

6.1 Introduction
6.2 Review of literature
6.3 Overall System Structure of the Rapid and Economic Product
   Development
6.4 Cost Index Structure
6.5 Cost Estimation and Optimisation
6.5.1 Generative Cost Estimate Method
6.5.2 Variant and Knowledge based Cost Estimate Method
6.5.3 Some Considerations on Cost Estimate
6.6 Implementation of the Cost Estimate and Optimal Control System
6.6.1 Case Scenario and System Description
6.6.2 Cost Estimate and Optimization for Making the Case Part
6.7 Chapter Conclusions
CHAPTER 7
AN INTEGRATED DATA STRUCTURE FOR SUPPORTING RAPID PRODUCT DEVELOPMENT
7.1 Introduction
7.2 Overview of System Structure
7.3 STEP-Based Integrated Data Structure
7.4 The WWW based Knowledge base System
7.4.1 Knowledge base structure
7.4.2 Industrial Implementation
7.5 Chapter Conclusions
CHAPTER 8
AN INTERNET BASED PRODUCT INFORMATION MANAGEMENT SYSTEM FOR RAPID AND INTEGRATED MOULD PRODUCT DEVELOPMENT
8.1 Introduction
8.2 The System Framework
8.3 The Internet-based information Management System
8.3.1 Product Data Model
8.3.2 Product Information Management System
8.3.2.1 OODBMS Structure
8.3.2.2 STEP Database Structure
8.3.3 WWW Database Tool
8.4. The Internet-based System Structure
8.4.1 IAT and CCT
8.4.2 Incremental Process Planning (IPP)
8.4.3 Cost Optimisation Model
8.5 System Implementation
8.6 Conclusion of the Chapter

CHAPTER 9
INTERNET BASED DFX FOR RAPID TOOL/MOULD MAKING
9.1 Introduction
9.2 System Structure
9.3 DFX Knowledge Base Structure
  9.3.1 Knowledge Base Structure
  9.3.2 Generic DFX Model
9.4 DFX via the Internet
  9.4.1 IDFM for Rapid Development of Moulds
  9.4.2 IDFC for Rapid Development of Moulds
9.5 Chapter Conclusion

CHAPTER 10
CONCLUSIONS AND FUTURE WORK
10.1 The Goal of This Thesis Work and its facing problems
10.2 Major Achievements and Research Findings
10.3 Future Work

APPENDIX
REFERENCES
VITA
LIST OF TABLES

Table 1.1- Source: Product Development Performance: Strategy, Organization, and Management in the World Auto Industry. 5
Table 5.1- Algorithm Comparisons 69
Table 9.1. Costs for the manufacture of the injection mould 126
LIST OF FIGURES

Figure 1.1 - RPD Metrics and Perceptions 2
Figure 1.2 - Financial and lead-time Results 4
Figure 1.3 - The structure of the Thesis 10
Figure 2.1 - An integrated environment for developing IRPD systems 28
Figure 2.2 - The structure of an IRPD platform for rapid mould product development 35
Figure 2.3 - The structure of the IRPD environment 36
Figure 3.1 - The integrated product development system for sheet metal concurrent design and manufacturing 40
Figure 3.2 - A selection of entities from the EXPRESS schema for the knowledge bases design 45
Figure 3.3 - Case studies for the integrated product development system 47
Figure 4.1 - The role of the information framework in the integrated product development system 50
Figure 4.2 - Data translators versus integrated data environment 51
Figure 4.3 - The position of the middle property layer 52
Figure 4.4 - A generic handling method of bend sheet metal part 52
Figure 4.5 - The information integration framework of sheet metal parts 54
Figure 4.6 - An example for building feature relationships graph for a sheet metal part 55
Figure 4.7 - The tree structure of shape features 55
Figure 4.8 - The two branches structure of shape features 56
Figure 4.9 - A common information model of sheet metal parts 56
Figure 4.10 - An example sheet metal parts for object-oriented information model 57
Figure 4.11 - The Object-oriented information expression of the sheet metal part 58
Figure 5.1 - A sample sheet metal part for compound manufacturing 60
Figure 5.2 - The CAD/CAPP/CAM integrated system structure for sheet metal compound manufacturing 62
Figure 5.3 - Object-Oriented expression of sheet metal parts 64
Figure 5.4 - Object-oriented expression of inner holes feature - tool chain list 64
Figure 5.5 - The data structure and functions of integrated CAPP system for 65
sheet metal compound manufacturing

Figure 5.6 -Tools-Tools position chain-list  66
Figure 5.7 -“cut-in” and “cut-out” arc for outer contour cutting  70
Figure 5.8 -“cut- out” and “cut-in” arc for inner contour cutting  70
Figure 5.9 -Circular auxiliary paths for outer contour manufacturing  71
Figure 5.10 -A simulation result for cutting, punching and compound manufacturing  74

Figure 6.1 -The overall structure of the rapid and economic product development system  80
Figure 6.2 -Cost Index Structure  84
Figure 6.3 -A Costing Knowledge base for sheet metal manufacturing  85
Figure 6.4 -Case part  89
Figure 6.5 -A screen printout from the Manufacturing Process Planning and optimization System for sheet metal parts  90
Figure 6.6 -Cost Index Structure of the case part  91
Figure 6.7 -Cost Index Structure that includes alternative processes  92
Figure 6.8 -Dynamic programming network for making a part through alternative operation routines  93
Figure 7.1 -The Structure of the WWW based rapid mould product development system  96
Figure 7.2 -The home page of the RPD information management system  97
Figure 7.3 -Data structure of the design/manufacturing knowledge base  99
Figure 7.4 -The integrated product data model in EXPRESS-G for an injection mould  99
Figure 7.5 -The structure of the WWW based information management system  101
Figure 7.6 -The product interface  102
Figure 7.7 -The tool interface  103
Figure 7.8 General information on surface finish requirements of mould M-5112  104
Figure 8.1 -Framework of the WWW-based integrated product development information management system.  108
Figure 8.2 -Distributed information structure for a plastic mould design and manufacturing.  109
Figure 8.3 -The structure of an OODBMS for mould product development  110
Figure 8.4 -A knowledge base example for mould/tool product  111
design/manufacturing

Figure 8.5 - The WWW database tool (WDT) developed by using multi-computer languages 112

Figure 8.6 - The structure of the WWW-based integrated information management system 113

Figure 8.7 - Information flows among the different models 114

Figure 8.8 - WWW-based integrated information management system for mould products design and manufacturing 116

Figure 9.1 - The structure of the Internet based DFX system 119

Figure 9.2 - DFM Knowledge bases structure 120

Figure 9.3 - DFM via the Internet 122

Figure 9.4 - Basic flow chart for IDFC of Plastic Injection Moulds 124

Figure 9.5 - DFC via the Internet 125

Figure 9.6 - Mode of the two-cavity, two-plate mould for front & rear panels 126
NOMENCLATURE

OKP-One-of-a-Kind Production
PD- Product Development
RPD-Rapid Product Development
CAD- Computer Aided Design
CAPP- Computer Aided Process Planning
CAM- Computer Aided Manufacturing
CE- Concurrent Engineering
DFX- Design For X
DFM- Design For Manufacturing
DFC- Design For Cost
GUIs- Graphical user interfaces
QFD- Quality Function Deployment
UoF- Units of Functionality
HoQ- Houses of Quality
VoC- Voice of Customer
IDFX- Internet based DFX
IDFC- Internet based DFC
IDFM- Internet based DFM
WWW- World Wide Web
CNC- Computer Number Control
DCOM- Distributed Component Object Model
SDAI- STEP Data Access Interface
CORBA- Common Object Request Broker Architecture
STEP- Standard for the Exchange of Product Data
ORBs- Object Request Brokers
CGI- Common Gateway Interface
WDT- WWW Database Tool
IAT- Information Access Tool
IPP- Incremental Process Planning
CCT- Collaborative Communication Tool
OODBMS- Object Oriented Database Management System
SE- Simultaneous Engineering
CSCW- Computer Supported Cooperative Work
SUMMARY

The globalization of business and industry and the worldwide competitive economy are forcing One-of-a-Kind Production (OKP) enterprises to utilize fully the best available equipment and techniques. The objective is to have efficient control of the organizational structure in order to produce high quality products at lower prices within a shorter period of time. With the development of modern technologies and theories, a new generation of OKP systems is urgently required for Rapid Product Development (RPD) by OKP enterprises to meet today's increasingly global competition.

RPD has been recognized as playing a key role in improving the competitiveness of OKP companies in the global market. With rapid development of information technology (especially Internet technology), network technology, modern management technology and other related advanced technologies to the manufacturing enterprises, tremendous efforts have been made in developing Internet based systems for the purpose of RPD through collaboration over the Internet, e.g. Internet-based systems for product design and manufacturing, Internet based design for X and Internet based integrated systems. These Internet based systems can be used to rapidly produce high quality products with low costs, and thus bring tremendous profit for manufacturing companies and thus stay competitive in the global market. Such kinds of systems are urgently required in industry. However, the research and development of Internet based RPD (IRPD) systems for producing OKP products are still in early stage. Further research and development in this area are becoming more and more urgent.

This thesis systematically reviews the historical background of the innovative technologies for Internet based OKP product development. Through this systematical overview of the existing Internet based systems and recent approaches of Internet based design and manufacturing systems, the problems that emerged from recent approaches have been identified. They include, distributed system structure, cost optimization algorithm, product data modeling, production information flow modeling, DFX, CAD/CAPP/CAM integration technology and process modeling and distributed knowledge management.

To overcome these problems and to develop a new generation of OKP systems, research and development of a product development platform has been carried out throughout this research work. The main findings are outlined in chapters, which include the definition of the proposed product development platform, an integrated information framework for rapid development of sheet metal parts, the global data structure for global manufacturing, a CAD/CAPP/CAM integration system, a WWW based information management system, cases studies of using the platform for rapid sheet metal and mould product development, the definition and application of an Internet based cost estimation and optimization framework. These research topics and findings constitute the main contents of the thesis.

Finally, the future trend of RPD has been proposed in this thesis, which covers the topics of Internet based collaborative design, decision support, manufacturing support, supply chain management, Internet techniques for product design and manufacturing, system integration, product modeling, STEP based data environment, concurrent engineering (CE), etc., which might be used to guide coming research, or to be used as reference for a company to deploy or develop Internet based RPD systems for producing OKP products.

During the course of this work, the following papers have been published in international journals and international conferences:


CHAPTER 1

BACKGROUND AND MOTIVATION

1.1 Rapid One-Of-a-Kind Product Development

Nowadays more and more manufacturing companies have realised that the ability to quickly develop a customised product through an economic and efficient way is critical for them to survive in the keen competitive international market, particularly for those one-of-a-kind production (OKP) companies.

Wortmann (1991) defined various types of one-of-a-kind production (OKP) by a two dimensions typology. One dimension is determined by a company's production system position strategies, i.e. a product oriented or a capacity oriented. The other dimension is determined by a company's market strategies, i.e. make to stock, assembly to order, make to order, and engineer to order. The research as reported in this thesis focuses on the rapid product development problems in the OKP company which has a product oriented production system and adopts an engineer to order market strategy.

OKP, as predicted by scholars like Rolstadas (1991), Wortmann (1991) and Hirsch (1992), could be a promising manufacturing model for the factory of the future. At the same time OKP poses some challenges to the factory of the future. Tu (1996) characterised the OKP philosophy as: 1) high customisation, 2) successful product development and production in one go, 3) optimal or rational utilisation of technologies and resources, 4) adaptive production planning and control, 5) continuous customer influence throughout the production, 6) incremental process planning, 7) distributed control and inter-organisational autonomy, and 8) virtual company structure and global manufacturing. From a practical viewpoint, an OKP company can be loosely understood as an advanced manufacturing company, which provides customised products in a certain manufacturing area (i.e. 'Kind' in OKP). It is somewhere between a capacity-focused job shop and a product-focused flow-line. According to its market needs, it may go to one end as a job shop to produce products in a relatively narrower product domain (compared with a capacity-focused job shop) or to the other end as a batch production manufacturing company. A typical OKP company can be a sheet metal company, injection mould/tool manufacturer, a steel construction company, etc. An OKP company has a higher flexibility to adopt the market changes than a mass/batch production company has, and higher production efficiency than a job shop has. However, in OKP companies, owing to a high customisation, involving large amount of uncertainties and consequently resulting in a lot of reworks, the product development cost is normally higher and the development lead time is longer than in those product-focused manufacturing companies.

To rapidly develop products with low costs, a new paradigm called Rapid Product Development (RPD) has been proposed for the purpose of rapid development of products with lower cost and acceptable quality. Bullinger et al. (2000) defined RPD as an interdisciplinary methodology to combine all influences of an engineering process to an iterative product development. Its research topics focus not only on product, but also their development process. The Rapid Development of OKP product is a holistic organizational concept that describes rapid OKP product development process achieved by combining and integrating various innovative technologies and tools, e.g. Rapid Manufacturing (http://www.albright1.com/Manufacture.shtml), simultaneous engineering (SE), CSCW (Computer Supported Cooperative Work) tools (Bullinger et al. 1996), and a supportive environment (Tu et al. 2001).

In this thesis, the RPD will only focus on the development of customized, individual or engineer to order OKP products. The 'P' in 'OKP' means OKP products, e.g. a customised sheet metal product or an injection mould. As shown in Figure 1.1, the RPD processes include all the OKP product development processes starting from the customer's requirements. The objectives of the RPD are:

1. To shorten OKP time to market (from product definition to market launch).
2. To develop OKP products by optimising the key factors, e.g. time to market, cost and quality.

3. To increase OKP products quality and decrease waste (changes, reworks and errors).

4. To rapidly respond to the customer’s requirements and market changes.

1.1.1 The Performance of RPD

The effectiveness of an OKP company’s product development process must be evaluated against some objective criteria - a set of quantifiable parameters - metrics (as shown in Figure 1.1). Generally speaking, the metrics typically fall into following four broad categories - quality, time, financial, and waste. Hence, the performance of product development can be weighed by following these factors. (Floyd 1993)

1) Product Quality

The definition of quality is often used in today’s manufacturing environment as “meeting the needs and expectations of customers.” In today’s market, it is no longer sufficient to define quality simply as “conformance to specifications”. Sometimes, the specification may not fully meet the customers’ needs that change with time. In the new definition, the operative words are needs, expectations and customers. The definitions of needs and expectations given by (Floyd 1993), are briefly quoted in the following:

Needs are those functions, characteristics, and features that customers expressly want or must have in a product that they are acquiring. Customers are aware of their needs and, if asked, can usually state them with a reasonable degree of clarity. Expectations are more difficult to determine and identify. Expectations are those things customers automatically assume to be inherent in the product. Often, they do not explicitly state their expectations.

2) Lead Time

There are many ways to define the time to develop an OKP product. Some common terms are Time to market, Development Lead Time, Product Development (PD) Time, and PD Cycle Time. These terms are generally used interchangeably and all intended to describe the total time it takes to implement the PD Process from the conception of a product idea to the point at which the product is both released for sale and in full production. In this thesis, the lead time is understood as the time from staring a product definition to finally delivering the product to the customer.

World-class OKP companies are constantly striving to improve their performance, and thereby to reduce the PD cycle time. Through a combination of innovative RPD process changes and continual improvement with new technology, many manufacturing companies have made dramatic reductions in the PD lead time. For example, during the past decades, companies striving for world-class status have cut cycle times as much as 50% as reviewed by (Floyd 1993). These companies are continuing to improve.
3) Financial

For many OKP companies, product development not only requires a significant amount of money; it also establishes the costs of producing OKP products. The Return on Investment (ROI) has always been on the focus of product development. In recent years, cost control and containment have become even more important than before.

PD Process costs are generally grouped into product development costs and product manufacturing costs. Product development costs are the total costs incurred through a product development process. These costs are frequently referred to as NRE (non-recurring engineering/development) costs since they only happen once. The cost for an OKP product is mainly PD cost or NRE cost since an OKP product will be developed in one-go and rarely repeated in future.

The manufacturing cost (sometimes called the recurring cost) represents the cost of producing one unit of the product. It includes material, labour and manufacturing overhead costs for the product. It is important to note that much of this cost is established during the PD process because of the design of the parts, the manufacturing methods, processes, tooling, etc. An OKP product also includes manufacturing cost as a part of its total development cost.

4) PD Waste

There are many contributors to waste in the RPD process, but the most costly factors are errors, reworks and changes. These result from lack of proper preparation and planning, poor implementation, and a poor PD process that allows development to proceed between responsible departments without sufficient information interchange. Designers, for example, may develop the product without taking into account the needs of the manufacturing departments or their departments. A flood of design changes may overwhelm the organization as it tries to modify the product to suit the manufacturing requirements. Although the cost of the changes may be huge, the biggest impact is the delay of market launch while the product is debugged. Another major source of waste results from inadequate product definition. The product designers may be required to rework many man-months of effort when the product definition is changed late in the development cycle.

1.1.2 Evaluating The PD Process

The first step in improving a PD Process is to measure the effectiveness of the one currently in use. This can be done by a comprehensive assessment that includes the use of both metrics and perceptions. The metrics are used as a quantitative assessment of the PD Process and the perceptions are used as a qualitative assessment of the process. The combination of the two results in a detailed evaluation. Metrics are used to determine quantitatively how well the process is meeting its objective of yielding timely, competitive, and profitable OKP products. Metrics can provide a basis for comparing the PD performance of a company with its competitors, as well as measuring the ability of the PD organization to develop products that meet their original plans and objectives.

The effectiveness of a PD Process requires its use and support by the customers of the process. If the users of the PD Process perceive that the employing process is not a quality one - it is very bureaucratic and cumbersome - then they will resist using this process. The process will be ignored and the product developers will proceed on their own way.

Management’s perception of the relative degree of success demonstrated may be critical to the funding of future OKP product development programs. If their perception is that product developments are consistently marginal, the probability of obtaining adequate funding for future programs becomes questionable. Management will have to be convinced, and rightfully so that the sins of the past have been addressed and are on the way to being corrected, before they can invest in the development of the product.

Thus, many companies find themselves in a situation where new products are not returning adequate
ROI to justify continuing long-term investment and without new products there is no hope for long-term survival. The solution would be to develop an effective PD process that will yield competitive and timely products.

1.1.3 Benefits of RPD

In the market, usually successful products achieve high marks in each of the categories of performance metrics, e.g. short lead time, low waste and low cost. Many of the metrics affect each other. Quality products, for example, can be sold well in the market, and hence lead to a higher life cycle profit.

Figure 1.2 demonstrates the benefits gained by the manufacturing industry through rapidly delivering quality products to the market. Obviously, company A entered its market early with a quality product, achieved a commanding market share, and generated much more life cycle revenue than its competitor, company B, who entered the market late with a product of equivalent quality.

Life cycle profits are negative during the PD cycle time and only turn positive when sufficient revenues are achieved. Figure 1.2 (3) depicts the corresponding product life cycle profits for companies A and B.

1.1.4 Drivers for RPD

There are many factors driving the need for OKP companies to improve their PD Process. Some of these factors are within the control of the company and some are not. Some factors within the control or influence of the company are:

1) ROI targets.
2) Staff reductions resulting from downsizing activities.
3) Excess production capacity.

These factors are specific to the company and its internal environment. They all obviously relate to increasing profitability and hence long term viability.

In addition to the company internal factors, there are several common external factors that are major drivers to force the company to move toward RPD. These factors include: 1) Aggressive global competition, 2) Rapidly changing technologies, and 3) Increasingly complex markets. These factors are briefly discussed in the following.

1) Aggressive Global Competition

Table 1.1 illustrates some key results during the global competition for the auto industry in the 1980s. Twenty auto manufacturers worldwide were evaluated and pertinent metrics determined. The metrics were normalized for a compact car. The chart clearly shows, as most of us already know -that the Japanese developed automobiles in far less time and for much fewer development dollars. The Japanese products were also better performers than those of the European high volume manufacturers.
<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>JAPANESE VOLUME PRODUCER</th>
<th>U.S. VOLUME PRODUCER</th>
<th>EUROPEAN VOLUME PRODUCER</th>
<th>EUROPEAN HIGH-END SPECIALISTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGINEERING HOURS (millions)</td>
<td>1.7</td>
<td>3.2</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>PERCENT TO US</td>
<td>53%</td>
<td>100%</td>
<td>94%</td>
<td>94%</td>
</tr>
<tr>
<td>LEAD TIME - CONCEPT TO MARKET (months)</td>
<td>45</td>
<td>60</td>
<td>57</td>
<td>63</td>
</tr>
<tr>
<td>PERCENT TO US</td>
<td>75%</td>
<td>100%</td>
<td>95%</td>
<td>105%</td>
</tr>
</tbody>
</table>


Although this analysis is for one industry, it is very representative of what is going on in the world. In this increasingly fierce competitive global economy, there are foreign as well as domestic competitors who can do it better, and, when they do, they will seize the market share. Virtually no industry or marketplace can be immune from these competitive challenges.

2) Rapid Changing Technologies

Previously, technology usually remained the same for several generations of the product. But this is not the case today. Today, many technology changes take place within each generation. In some high tech industries, such as the computer industry and the communication industry, it is not uncommon for technologies to change dramatically every two or three years.

As a result, product life cycles are rapidly declining. This, in turn, reduces life cycle revenues and profits. It is becoming more and more difficult to achieve satisfactory ROI, considering the expense of developing OKP products and the short periods available for obtaining financial recovery on that investment. Hence, companies are being forced to develop OKP products at a faster and faster pace. OKP companies that are not making significant strides in reducing their RPD cycle times are struggling to remain competitive.

3) Increasingly Complex Market

Markets continue to become more diverse and complex. Customers, increasingly exposed to the advertising and marketing efforts of global competitors, are becoming more sophisticated and demanding. Competitors, seeking a marketing niche, are decreasing prices and adding product differentiation. This increasingly complex market has put a lot of pressure on OKP manufacturing enterprises to meet the diverse customer needs and demands.

1.1.5 Emerging Research Issues

To improve the performance of OKP manufacturing enterprises and to win the RPD competition, the major research problems for OKP companies to compete in today's global market have been identified (Tu et al. 2002a):

1) High customisation

It is obvious that engineer to order will result in a high customisation. High customisation often leads to diversified customer's requirements and hence great uncertainties. These uncertainties give a company numerous rework, longer product development and production lead time, and high cost.
2) ‘Once’ successful approach

The production in OKP companies is different from that in batch or mass production companies. In a
batch or mass production company, a prototype of the product will be first made to approve the design and
to study the manufacturing processes. In OKP companies, no prototype will be made. The product must be
made in OKP companies in one-go, or a ‘once’ successful approach. It is not economic for a mould/tool
maker to make a prototype of the mould first and then to make the final mould. It would be ridiculous for a
large oil cargo building company to make a prototype of the ship to approve the design and study the
manufacturing processes first, and then to make another ship as the final product. They all must make their
products according to customer’s requirements in one-go. This ‘once’ successful approach of product
production in OKP companies gives a lot of challenges or problems for the companies, (a) uncertainties in
product design, process planning and shop floor scheduling, (b) owing to aforementioned uncertainties
product design and production must be carried out concurrently, or jointly planned and controlled through
the whole product development life cycle, (c) hidden costs or reworks caused by inappropriate or wrong
designs, process plan or production schedules, (e) it is difficult (if not impossible) to get a well planned
production schedule and inventory plan, and the OKP companies are often running in a fire fighting
situation.

3) Looser or Fatter Production Planning and Control

Owing to the uncertainties mentioned above, the OKP companies normally plan their production and
allocate production resources in a much looser or fatter way compared with what happens in those batch and
mass production companies. In a batch or mass manufacturing company, the production is normally
organised along a production line that synchronously moves according to a pre-defined cycle time. In a car
manufacturing company, for instance, the cycle time may be one to two minutes. This means the production
in this car manufacturing company can be synchronised by this cycle time. Operations in every section
through the whole plant will be able to be completed in just one to two minutes. In every cycle time, a
product (or a car) can be delivered out from the company’s final assembly line. This would be impossible
for an OKP company. In OKP companies, the production is asynchronous. The operations are loosely
planned according to rough estimates of the processes. This loosely planned asynchronous production
results in jumble material flows in an OKP company, hence a higher work in process inventory and cost.

4) Continual customer influence

In OKP, customers may change their requirements through the whole production procedure, which is
different from the batch and mass production where the customer’s requirements are confirmed or fixed
before the product design and manufacture. In OKP, owing to his limited knowledge on how to produce the
required product, the customer may change the requirements after the product has been partially made, e.g.
when the customer sees the design of the product, a plastic part made by rapid prototyping machine, a part
of the product, etc. In a worse case, a customer may wish to change some of his/her requirements or to add
some modifications in later stages of production. It is obvious that to be able to cope with this type of
continual customer influence will improve an OKP company’s market place.

5) Complicated product data and information flows

An OKP company may produce a number of products at the same time. Owing to the high
customisation of these products, each of these products must be managed individually. It is not rare for a
part of product A to be wrongly assembled into product B, a wrong order be placed to the supplier, a wrong
inventory kept in the company, etc. The information and data of product design, process planning, inventory
plan and control, and production schedule are made for each of the products and flow simultaneously in the
company. On the other hand, no OKP company, particularly those small and middle sized, would like to
equip itself as a super company which can meet all kinds of customer’s requirements. They often use a lot of
partners or subcontractors to compensate for their lack of manufacturing ability. This leads to complicated
product data and information flows not only within the OKP company but also across the companies.
6) Complicated logistics management

This is another problem resulting from the high customisation of OKP. After the discussion of problem 5), this becomes an obvious problem. The complexity of logistics management in an OKP company can be viewed internally and externally. Internally, an OKP company normally has a complicated material handling and inventory control. Externally, it may have an extensive outsource network that consists of complicated supplier chains and partner/subcontractor chains, and a customer network that consists of complicated delivery chains and sale chains.

1.2 Motivation

To solve the problems as mentioned above, OKP companies have been actively adopting new technologies, particularly those computer technologies, such as intra/internet communication technology, CAD/CAPP/CAM, rapid prototyping, MRPII/ERP (manufacturing resource planning/enterprise resource planning), PDM (product data management), and computer simulation technology. According to a survey (Shaw 2000), these technologies are also widely applied in New Zealand manufacturing companies. Despite the great contributions that have been made to develop in these areas (a comprehensive review of literature will be given in Chapter 2), these technologies are often used separately rather than in a systematic and integrated way. The applications of these technologies are lacking in system integration and synergy.

To help the manufacturing companies, particularly OKP companies, to rationally adopt these computer application technologies to effectively shorten their product development cycle time and reduce the product development cost, a research program has been funded by the Foundation of Science and Technology (FRST) of New Zealand (NZ) since 1998. This research program aims to provide these NZ OKP companies with a reference production management system framework, concepts and software tools to support the applications of these computer technologies. The research works as reported in this thesis are part of this research program, which mainly focus on the strategies, algorithms and technology for rapid development of OKP products.

The principal goal of this thesis work is to develop new technology, algorithms, tools and systems to reduce PD lead time and cut down cost for OKP product development. This principal goal is to be achieved by proposing an Internet based product development system with the integration of modern technologies such as Information Technology (IT), Rapid Product Development (RPD) paradigm, information modelling, distributed multi-objective optimisation, and Internet based information management. The developed Internet based RPD or IRPD system is for the strategic purpose of improving the New Zealand manufacturing level in the world, particularly for new product development. In the thesis, the theories, concepts and methods from the areas of modern control engineering, rapid manufacturing, CAD/CAPP/CAM integration, engineering optimisation, information technology and knowledge-based technology has developed and applied in industry. The research has used two typical New Zealand OKP companies, i.e. customized sheet metal manufacturers and tool/mould makers, as pilot companies or test beds to test the proposed system and other major research findings. Following similar principles, it can be envisaged that a number of high-tech RPD tools can be developed to help other manufacturing enterprises in New Zealand compete in the world market

1.3 Thesis objectives

The presentation of this thesis strives for a balance between modularity and integrity of the chapters. Chapters 3 through 9 of the thesis are rewritten from the published journal or conference papers and are carefully structured in a self-contained fashion. Each of these chapters starts with an overview of the technique and research background of the major objectives of the research, which form the main points of the IRPD system structure, concept and method. The major objectives of the thesis include:

1) Investigation of current hurdles that hamper RPD and proposing reference architecture for building an Internet based integrated product development system.
Nowadays, the operation of an OKP company tends to build on the basis of co-operation between the company and its partners in its manufacturing network. It is often driven by customer focus and volatile markets. In short, OKP companies are running in a global competitive and collaborative environment. To rapidly develop a new product in this global environment, the OKP companies need to update their traditional manufacturing systems and traditional product development processes to meet the requirements of this global environment. The hurdles to block the way of this upgrading process need to be identified and further studied. To meet the requirements of the global environment and to facilitate the complicated communications between an OKP company and its partners and between the company and its customers, a reference structure of an Internet based integrated product development system needs to be developed.

2) Applying the proposing system architecture for rapid sheet metal product development

To test the feasibility of the proposed reference system architecture, the proposed system architecture needs to be implemented to speed up a sheet metal product development. Sheet metal manufacture is the main part of the New Zealand manufacturing sector.

3) Developing an Internet based information platform or framework for rapid sheet metal part/product development

To rapidly develop a sheet metal part/product, an Internet based information platform needs to be further developed as a major part of the Internet based rapid sheet metal part/product development system.

4) Developing a compound sheet metal cutting and punching method to speed up the sheet metal product development

A compound cutting and punching production method supported by an integrated CAD/CAPP/CAM system in sheet metal manufacturing needs to be developed to speed up the sheet metal product design and manufacture. Many existing commercial CAD/CAM systems are not suitable for this manufacturing method, especially under concurrent and global design and manufacturing environments. Some problems have to be solved before these CAD/CAM systems can be employed and integrated for this compound manufacturing method.

5) Cost estimation and optimisation for rapid development of sheet metal part/product

In order to keep a strong competitive ability, sheet metal manufacturing companies always have to look for the cheapest way to develop their products. This is in fact a cost estimation and optimisation process.

6) Proposing an integrated product data model for product development life cycle and applying it for rapid tool/mould making.

As mentioned before, OKP companies tend to design, manufacture and maintain a product through global collaboration in a team. The team members may include engineering, manufacturing and service firms and they are often geographically dispersed. They work together to design, manufacture and support the product. For this reason, an integrated information model has to be developed to support the definition and exchange of the product data unambiguously and consistently between the team members during the entire course of the product development process. Developing and manufacturing a product through this type of global collaboration is termed virtual manufacturing, and a company that adopts the virtual manufacturing strategy is also called the virtual manufacturing company. It is obvious that the virtual manufacturing concept is particularly suitable for rapid OKP product development since the virtual manufacturing strategy can improve the company's flexibility and capacity to handle the much wider and diversified customer requirements. Tool/mould making is a typical one of a kind production and it is also another major part of the New Zealand manufacturing sector. To apply the research findings to widely support the New Zealand manufacturing companies, the proposed data model is going to be tested through a mould development instead of a sheet metal part/product development.
7) Developing an Internet based information management system

It is obvious that an Internet based information management system needs to be developed as an important part of the Internet based OKP product development system to manage the various databases which can be directly accessed via Internets.

8) Proposing an Internet based DFX (IDFX) system for rapidly tool/mould making

DFX, or Design For X, is a broader concept to encapsulate various concurrent engineering approaches for RPD, such as design for manufacture (DFM), design for assembly (DFA), etc. To test the suitability and feasibility of adopting this concept into the Internet based rapid OKP product development system, an Internet based DFX system or IDFX system needs to be developed and tested in a typical New Zealand OKP company.

1.4 Overview of the Thesis

In line with the thesis objectives, the thesis is structured into ten chapters, which are briefly indexed and illustrated in Figure 1.3. Chapters 2 through 10 are outlined in the following:

Chapter 2- The State-of-the-art Literature Review and a proposed Reference System Architecture

In this chapter, the author systematically overviews the historical background of Internet based OKP product design and manufacturing systems. Through systematically overviews the existing OKP systems and recent approaches of Internet based design and manufacturing systems, the author discusses the current techniques that can be used to implement next generation OKP systems, i.e. Internet based integrated product development systems. The problems that have emerged from recent developments have been identified. Future trends and some relevant new technologies are also identified, which cover the topics of Internet based collaborative design, decision support, manufacturing support, supply chain management, workflow management, Internet techniques for product design and manufacturing, product modelling, STEP based data environment, concurrent engineering, etc. These relevant technologies may be used to guide the coming research or to be used as references for a company to implement IRPD systems. Lastly, at the end of this chapter, a reference system architecture for building an Internet based integrated product development system has been proposed for Internet based rapid development of OKP products.

Chapter 3- A Internet based Integrated Product Development System for Rapid Sheet Metal Part/Product Development

In this chapter, the application of reference system architecture as proposed in Chapter 2 for rapid sheet metal part/product development is addressed. Several major modules of the system are discussed. They include an integrated product development software platform, an information integration framework, a RTCAPP (real time computer aided process planning) module and design/manufacturing knowledge bases. The principles and work processes of these modules are demonstrated through industrial case studies.

Chapter 4- An Information Framework for Rapid Sheet Metal Part/Product Development

This chapter explores the definition and the structure of an information framework for concurrent design and manufacturing of a sheet metal part/product. This framework aims to build an information bridge to fill the gap between the sheet metal part design, process planning and manufacturing systems. It is based on the principles, which are called zero thickness and zero bead radius. These ‘zero’ principles are used to abstract the geometry entities of sheet metal parts in order to facilitate part modelling and information modelling. In this chapter, a tree based step-structure information modelling methodology for sheet metal parts has been put forward and a case study has been given.
Figure 1.3 the structure of the thesis
Chapter 5 – Compound sheet metal cutting and punching method to speed up the sheet metal product development

This chapter presents a compound cutting and punching production method supported by an integrated CAD/CAPP/CAM system in sheet metal manufacturing. The solutions include an integrated data integration platform based on Pro/INTRALINK and STEP, and a knowledge-based real time CAPP (RTCAPP) system for compound sheet metal cutting and punching. Within the presented CAD/CAPP/CAM system, some key modules have been developed. They are the automatic tool selection and manufacturing sequencing module, a shortest tool path optimisation module, a cost estimation module and an automatic insertion of an auxiliary path module based on knowledge bases. These modules will be addressed in this chapter.

Chapter 6 - An Internet based Cost Estimation and Optimisation Framework for Sheet Metal Part/Product Development

In this chapter, an Internet based cost estimation and optimisation framework are developed to estimate and optimise the product life cycle cost through the tree representation of alternate development processes. Although, the life cycle cost considered in this paper is focused on design, delivery, manufacturing costs, the proposed framework also extends well for covering other sources of costs. To show the full scope of the system, the Internet based cost framework and the optimisation methodology will be discussed in detail. Unlike some earlier research that analyses a single arbitrary process sequence in a single manufacturing company, this Internet based cost estimation system developed considers the complexities of global manufacturing and contains an optimisation model for automatic selection of the best process sequence from among all feasible alternative sequences. The process sequence selection is based on the minimization of the total product development cost from among all the feasible process sequences. Because it is assumed that the process costs are sequence-dependent, a mathematical model combined with a heuristic solution is employed to determine the lowest cost estimate and consequently, the minimum cost process sequence can be achieved. By integrating the cost framework with the integrated product platform proposed by Tu and Xie (2000), an Internet based software platform will be developed to help global manufacturing enterprises develop products rapidly and economically.

Chapter 7 –An integrated global data structure for supporting rapid product development

This chapter reports a research work that aims at developing an integrated data structure to support the rapid product development (RPD). The emphasis is placed on the integrated data management and knowledge reuse to shorten the product development cycle. The integrated data structure is modelled by the EXPRESS of STEP. In terms of this data structure, a design/manufacturing knowledge base was developed as a major part of the WWW (world wide web) based product development system. The basic principles and concepts of the knowledge base and the WWW based information management system will be presented in the chapter. The feasibility of the developed product data model is tested through an injection mould development cycle.

Chapter 8 -An Internet-based Product Information Management System

This chapter explores the design and development of a World Wide Web (WWW) based product development information management system for a cross-nation manufacturing corporation that is headed by a holding company in Christchurch, New Zealand. Since product data are often managed in a distributed computing environment, common object request broker architecture (CORBA) is employed to ensure the interoperability among distributed information objects. This WWW based information management system includes two major components: 1) WWW based product design and development distributed object oriented databases and knowledge bases; 2) A WWW based integrated system platform. Several sub-models are introduced, which include an object oriented database structure, a WWW-based information
management system, a WWW database tool, an information access tool, and an integrated software platform for the integration of CAD, CAPP, CAM and WWW based information management system.

Chapter 9 - Internet Based DFX for Rapid Tool/Mould Making

The Internet offers tremendous potential for building computer communication and software platforms for rapid development of products to meet global competition. In the past several years, Internet based Design for X (DFX) has been recognized as playing a key role in improving the competitiveness of manufacturing companies as it can be used to rapidly produce high quality products with lower cost, and thus bring tremendous profit and make their products more competitive in the market. However, the implementation of DFX systems is not an easy task, the reason is that there are many new techniques involved in this field and the product design and manufacturing processes are normally sophisticated and vary with products, product designs and manufacturing environments. In this chapter, a new approach of Internet based DFX system is proposed for rapid development of plastic injection moulds. The structure and the key models of the system are discussed.

Chapter 10 – Conclusions and Future Work

The last chapter of the thesis concludes with the results obtained from this thesis work. First, a short summary is given, and then an outline of the results and contributions obtained during the research period are presented together with a discussion of their effect on the research objectives. Next, an evaluation of the thesis and the research is made and a number of overall questions concerning the development of an integrated product platform for RPD are discussed, thus evaluating the synergy of all results obtained. Finally, important aspects to be considered in future work are discussed.
CHAPTER 2

THE STATE-OF-THE-ART LITERATURE REVIEW AND A PROPOSED REFERENCE SYSTEM STRUCTURE

In this chapter, the historical background of Internet based product design and manufacturing systems for rapid development of One-of-a-Kind (OKP) products will be systematically overviewed. Through systematically over viewing the existing OKP systems and recent approaches of Internet based design and manufacturing systems, the author will discuss current techniques that can be used to implement Internet based product design and manufacturing systems for producing OKP products. The problems that emerged from recent developments are to be reviewed and sorted in this chapter. The future trends of Internet based collaborative design, decision support, manufacturing support, supply chain management, workflow management, Internet techniques for product design and manufacturing, product modelling, STEP based data environment, concurrent engineering, etc., will also be discussed in this Chapter. These reviewed state-of-the-art works have been used directly or indirectly as references for this thesis work. In the end of the chapter, a reference system structure for building an Internet based integrated product development system is proposed for rapid development of OKP products.

2.1 Introduction

Nowadays OKP manufacturing enterprises are facing more severe competition in the market than ever before owing to the globalisation of the market, short life cycle of products, increasing product diversity, high demand for quality and short delivery times. To survive in the market, these enterprises have to work hard to achieve high quality and productivity, reduce cost, and respond quickly and effectively to a market that is becoming more global, dynamic and customer-driven.

In OKP companies, owing to a high customisation, involving a large amount of uncertainties and consequently resulting in a lot of reworks, the product development cost is normally higher and the development lead-time is longer than those of product-focused manufacturing companies. To rapidly respond to market pressure, many OKP companies have adopted a virtual manufacturing concept, which can be simply understood as a global collaboration business through sub-contracting and technology transfer between a company and its partners. This virtual manufacturing concept helps a company to quickly and often economically meet its customers' needs by joining the manufacturing strengths of its own and its partners'. However, these virtual companies usually require support technically and theoretically to manage and optimise product development processes through a whole product development cycle. These processes include customer requirements interpretation to decide how to address the customers' needs, product design to address the customer's needs, conceptual and functional prototyping to prove the effectiveness of the designs, manufacturing process planning (which includes sub-contract planning) to determine how the product is to be made, design manufacture of tools, pre-production trial, production, and finally delivering the product to the customer or launching the product into the market. Those OKP companies, particularly wish to meet the customers' needs in one-go and minimize the costs spent on the intermediate tests and mock-up manufacture.

To shorten the lead-time and to reduce the cost of product development for OKP companies, particularly for those virtual OKP companies, the combination and integration of existing innovative...
Chapter 2 The State-of-the-art Literature Review and A Proposed Reference System

Structure

technologies and the development of new technologies have become essential. The achievement of the goals of RPD is becoming more and more difficult as the PD environment has dramatically changed and the factors that influence the PD process have become more and more complicated. The goals of RPD cannot be achieved by only focusing efforts on a single stage of the PD process, but need the contributions from the whole PD process. Hence, an Internet based PD system has to be built to support the PD process under a geographically distributed environment (Tu et al. 2000b). This system can be used to support and facilitate the communications and data flow at different stages through a PD lifecycle. To develop this PD system, new technologies have to be developed and integrated for improving the overall PD performance, e.g. new CAD technology, collaborative design technology, Internet communication technology, information technology, artificial intelligence (AI), knowledge base technology, cost/lead time optimisation technology and decision support technology. These new technologies, that can continually improve product design and manufacturing ability, have become more and more essential for today’s OKP manufacturing companies to effectively compete in the global market.

As an enabling technology, the Internet has been widely used in the PD field and can enable intelligent product design and agile manufacturing by updating traditional plant information systems and eliminate barriers to integration. It has launched a revolution in worldwide data transfer and communication. Its platform-independent architecture allows manufacturers to deploy applications across virtual organization, which includes business partners/customers whose computing environments may be completely different. In the PD field, the application of the Internet has been used in numerous areas including marketing, management, design, manufacturing, production, planning, customer service, etc. (Su et al. 2001). Internet based RPD (IRPD) has been recognized as an efficient way to help manufacturing companies (especially OKP companies) stay competitive in the global market (Bullinger et al 1996). However, there are problems that hamper the development of IRPD systems, e.g. the integration of various technologies and tools with different enterprise data models (Tu et al. 2001, Huang et al. 2001a), worldwide knowledge and resource access among heterogeneous computer systems, the concurrent architecture for execution (Schaefer 1994), the ‘seamless’ integration of the existing systems that are fragmented and nearly isolated employed at various stages of a whole PD process, information and knowledge sharing between various systems that are employed at different stages of a PD process (Xie et al. 2001a) and remote access to algorithms or large application tools that run in different platforms (Wagner et al. 1996). Hence, to meet the needs of rapid development of OKP products, existing RPD technology and systems have to be further developed in terms of communication, collaboration, applying new technologies and system integration, e.g. customer-oriented virtual manufacturing, distributed concurrent engineering, global optimisation and the integration of various functional systems at different stages of PD process. These issues have been in the focus of both scholars and practitioners.

Besides exploring and discussing the questions as mentioned above, this chapter will also place its emphases on reviewing the past research and to systemically figure out how current technologies have been used, and what are the future trends to implement the IRPD in today’s industrial environment. A comprehensive summary and review of historical literature and projection of future trends are helpful for OKP companies to make decisions concerning the implementation of Internet based systems or upgrading existing manufacturing systems. They are also very helpful in figuring out possible future research directions. For large companies that have their own research facilities, the work presented in this chapter can facilitate their development work and point out current issues and the right direction. For medium or small companies that are looking for Internet based systems, this work will help them in the selection of satisfactory Internet based systems to meet their specific PD environment. In this chapter, some current technologies are discussed. A table that lists some of the existing Internet based PD systems and research projects are presented in the Appendix I. The characteristics of the systems and recent research projects are listed. Also, this chapter reviews recent approaches and research in Internet based PD area and presents a systematical approach for implementing IRPD systems, which can help to resolve the issues mentioned above. A key research issue will be addressed, which is, in general: How to build an Internet based distributed integrated PD system to achieve the goal of ‘rapid development of OKP Products’.
Chapter 2 The State-of-the-art Literature Review and A Proposed Reference System
Structure

The chapter is structured as follows. Section 2.2 addresses the overview related literature of OKP and Internet based PD approaches. In section 2.3, the emerging research topics that exist in the implementation of Internet based systems are discussed. In section 2.4, the key technologies that can be employed for developing IRPD systems are reviewed. Lastly in Section 7, a reference system structure for building Internet based RPD systems is proposed.

2.2 Overview of OKP and Internet based Product Development Systems

Literature and research reports regarding OKP and Internet based product development systems are emerging in recent years. Here, from among a large amount of literature and reports, I selected some typical ones and discuss them briefly in this section from the following aspects, i.e. research focuses and trend of OKP, requirements for the next generation OKP systems and the role of the IRPD systems.

2.2.1 Review of OKP

The OKP was intensively researched under the ESPRIT basic research on the theory of the factory of the future (FOF) in late 1980s and early 1990s (Rolstadas 1991). Most of the research efforts have been placed on the OKP production management because operation efficiency is believed to be one of the major bottlenecks in OKP. Rolstadas (1991) proposed a framework for managing OKP, in which an adaptive scheme to change the production resource structure (as presented by a R-graph) to meet the changes of product structure (as presented by a P-graph) was suggested. However, the limitations and static nature of this adaptive factory managing structure were also noticed. Up to now, the research in OKP is still heavily focused on the managerial problems in OKP. Not much work has been carried out on how to rapidly and economically develop OKP products. William (1995) identified some interesting research topics in made-to-order products, which include cost estimation and design decision support through application of advanced computer technologies.

Customisation is commonly understood as a company's ability to satisfy various needs of customers by economically changing its product or service design and manufacture. It has been recognized as one of the main focuses in an agile manufacturing system (Kidd 1994). Meredith et al. (1998) believed that an ideal agile manufacturing company was able to make a product at the same unit cost of making it in a batch of 10,000 units. This has set a clear requirement for improving the production efficiency in OKP.

Mass customisation is another popular term that has been widely used to note the mass producing of a customised product or OKP. Tseng et al. (1998) characterized mass customisation as recognizing each customer as an individual, meanwhile extracting maximum commonality to achieve scale of economy. One of the main techniques to achieve mass customisation is to develop product families and link them with product functional trees, thus linking customer requirements with product families. Jiao et al. (1999) presented a product family data model to support mass customisation practices. Johan (1997) used functional trees to express design history information. However, both Jiao et al. and Johan did not take the downstream manufacturing applications into their considerations.

To support the customisation, wider research has taken place on the methods of product design and manufacture. The typical achievements among these research efforts would be the so-called design for modularity (Exixon 1996, Marshall et al. 1999), variant design (Fowler 1996, Mckay et al. 1996), design for manufacture (Gupta et al. 1997, Chen et al. 1998), design for production management (Nielsen et al. 1995, Yang et al 1999), and computer aided process planning (CAPP) (Alting et al. 1989, Wu et al. 1998, Tu et al. 2000a). Marshall et al. (1999) developed a systematic approach for product modularity. However, design for modularity solves only a part of the customisation problems, the developed modules by no means meet the entire customer needs, especially highly customized OKP products.

Variant design is a method to adapt existing design specifications to satisfy new design goals and constraints. This normally consists of two basic components, i.e. an extensive database to record product
families and the possible variations and a sound reasoning method (e.g., case based reasoning) to find solutions from the history data in the database. McKay et al. (1996) provided an improved data model to support product variety and pointed out the limitations of using EXPRESS in STEP (Standard for Exchange of Product model data) to model product variety.

Design for manufacture (DFM) places its emphasis on the integration or links between product design and manufacture. Gupta et al. (1997) widely reviewed the research work on the DFM and pointed out that the common features of various DFM methods were cross-functional teams, good/bad examples, feature based evaluation and empirical parametric evaluation. In the DFM approach, the automatic feature recognition and feature-based design method have been recognized as interesting research topics and widely studied in the world. The typical feature recognition methods include syntactic pattern recognition, state transition diagrams, decomposition method, set theoretic approach, graph based approach, and external access directions (Maropoomos 1995, Lin et al. 1997). Some successful applications of feature recognition and feature-based design have been found from the literature. Lee et al. (1999), for instance, proposed a feature-based approach to generate alternative interpretations of machining features from a feature-based design model. Lau et al. (1998) presented a method to recognize manufacturing features from a STEP 203 file. However, as pointed out by Shah et al. (1997), overlapped feature recognition is still an unsolved problem. In practice, the definition, identification and automatic recognition of design and manufacture features are the main problems that impede the application of feature-based design method.

A few research projects have addressed the problem of design for production management. Nielsen et al. (1995) presented an approach to separate product features and the feature interfaces thus to simplify the production control activities from a supply chain perspective. Yang et al. (1999) gave an example of integrating CAD system with MRP system based on STEP protocol. Hence, the design choices reduced the impact on the required materials. However, these works did not completely address the integration problems between the product design and process planning, and between process planning and shop floor scheduling, which are critical problems for rapid and economic development of OKP products.

To rapidly respond to a market chance, the efforts in integration of process planning and scheduling in OKP have drawn more and more interest and attention in last few years. Mamalis et al. (1996) presented an on-line integrated process planning and production scheduling system. In Mamalis’s work, simulation tools were used to support the integrated process planning and production planning tasks. Saygin (1999) also presented a method for integrated flexible process planning and scheduling. Howard et al. (1998) proposed a generic manufacturing planning and control system architecture for different manufacturing environments. However, this work was not concerned with the involved product design tasks. Aidakhialallah (1999) developed the architecture for concurrent product design, process planning, and production control. However, the approach oversimplified the problem domains.

Morgan et al. (1990) indicated that the enterprise integration could lead to the high agility of a manufacturing company. As suggested by Tu (1996), OKP companies need a higher agility to meet various customers’ needs and a higher integration to shorten the product development cycle and to save the product development cost. Lim et al. (1998) identified four integration domains: business integration, business data integration, software integration and computer system integration. Jim et al. (1999) classified system integration into four stacked layers, viz., manufacturing system layer, production and process layer, computer system layer, and communication system layer. Numerous research efforts have been made to develop integrated system architectures, e.g., AMRF (automated manufacturing research facility) developed by the National Institute of Standards Technology in USA, CIMOSA (open system architecture for CIM) (Kosanke 1991) in Europe, and Purdue architecture (Bernus et al. 1996). However, these integrated system architectures were developed according to a formal system hierarchy, which normally exists in a traditional product-focused or batch manufacturing company (Toh et al. 1998). Manufacturing systems, which are built according to these architectures, cannot handle the frequent system structure changes resulting from the customisation in OKP (Tu 1997b). To solve the conflict between the system flexibility and integration, distributed object technology of computer communication has been recognized as a promising solution. By
using this technology, various production systems are treated as clients. These clients can be freely plugged into or taken off a heterogeneous computer communication network (Orfali et al. 1999).

More recently, it has been widely believed that the computer intra-Inter-net communication technology would lead to a communication revolution in manufacturing. Recently, some research efforts have been made to develop WWW (world wide web) based manufacturing systems. Indrusiak (1998) carried out a so-called CAVE project, which aimed to develop a WWW-based design automation framework. The implementation and integration of some CAD tools according to the CAVE standards were addressed in their report (Indrusiak 1998). CORBA (Mowbray et al. 1997a) and DCOM (Li et al. 1997) have been used for the development and deployment of WWW-based systems integration in a distributed heterogeneous environment. Menzel et al. (1999) presented a hybrid system using distributed simulations objects and fuzzy set theory to improve the system efficiency and flexibility. Dorador et al. (1999) explored the definition of the structure of information models for supporting the related processes of design for assembly and assembly process planning to achieve the intensive share of information required in concurrent engineering through the life cycle of the product.

From the above literature review, it is obvious that the various methods and techniques for RPD have been widely researched. Some methods and techniques are well established and some others are still in early research stages. But, the research trend of OKP is clear, i.e. to develop a new generation of production systems by using computer aided and integrated technologies and some management concepts to economically and rapidly develop a customised product.

2.2.2 Requirements for the Next Generation of OKP Systems

In recent years, the manufacturing industry has experienced tremendous changes, e.g. from mass production through batch production, flexible manufacturing and lean manufacturing towards agile manufacturing or OKP philosophy (Booth 1996, Cheng et al. 2001, Tu 1997c). These changes are directly driven by various requirements for a low product cost, high quality, high performance, and customer’s choice and requirements, etc. They result from the unexpected changes of competitive market environment, e.g. globalisation of market, a variety of customers’ demands, customer-designed products, and shortened product life cycle. These market factors have great impact on all of the PD-related activities, e.g. product definition, design, manufacturing process planning, production, workshop floor control, quality control delivery, and marketing. To meet these changes and challenges, a new OKP system needs to be developed.

In today’s manufacturing practice, products are rarely designed, manufactured and maintained by a single company, especially in these small and medium sized (SMS) companies. An SMS company normally does not have the breadth of knowledge and the capability to understand all aspects of a PD process (Geller et al. 1995). Hence to develop strategic alliances and to run the business or manufacture with various “partners” who have the wanted knowledge/expertise and hardware/software resources has been widely adopted by SMS companies. This provides global competitive advantages for these SMS companies. This collaborative production strategy was well referred to as a virtual manufacturing concept. By adopting the virtual manufacturing concept, the manufacturing system environment thus becomes heterogeneous as partners may use different design/manufacture tools, quality inspection systems, control facilities and different production resources and technologies. Such a heterogeneous manufacturing system environment will inevitably result in problems in management, communication and production. It will also hamper the systematic upgrading of the design and manufacturing tools/facilities, as well as affect the competitiveness of a company (Kim et al. 1998, Feldmann and Gohring 1999). Hence, a new generation of OKP systems is required to cope with this heterogeneous manufacturing system environment.

As mentioned before in this chapter, the manufacturing market tends to be global and dynamic. New technologies are continually emerging, and product life cycle is getting shorter. Manufacturing strategies should therefore shift to support global competitiveness, new product innovation, and rapid market responsiveness. Hence a new generation of OKP systems should contain key features, e.g. agility and rapid
responsiveness, to maintain competitiveness in the global marketplace and the ability to rapidly combine the strengths of "partners" to meet market needs. It will thus be more time-oriented, while still focusing on cost and product quality. The fundamental requirements for a new OKP system are identified in the following:

(a) Enterprise integration: In order to support global competitiveness and rapid market responsiveness, an individual OKP company will have to integrate its PD processes (with its partners via networks).

(b) Organizations are globally distributed (Tu et al. 2000b): This includes distributed knowledge-base systems or product information systems that are needed to support PD processes.

(c) Being able to cope with the heterogeneous and distributed manufacturing system environments.

(d) Open and dynamic structure: the manufacturing systems should be able to dynamically integrate new subsystems into the environment for specific applications or remove existing systems from the systems without influencing the basic structure of the working environment. This requires open and dynamic system architecture.

(e) Support cooperation and collaboration: the manufacturing systems should support geographically distributed teamwork, which includes cooperation and collaboration among team members.

(f) Agility and high customisation: the manufacturing systems must be used to shorten the PD cycle time and to respond to customers' requirements quickly.

(g) Technical advancement: New OKP systems need to adopt new technologies that are developed for a specific stage of the PD process to keep its advantages over existing manufacturing systems.

(h) Compatible with most of existing PD software tools.

(i) Stable and easy to use and maintain: The system should be user-friendly and fault tolerant both at the system level and sub system level so as to detect and recover from system failures at any level and minimize their impact on the working environment.

2.2.3 Role of IRPD

To develop the production systems that can meet the requirements of the new generation of OKP systems as mentioned in previous section, various attempts have been made from the two aspects, i.e. product design and manufacture (Chui and Wright 1999, Reed and Ajieh 1998). The results have illustrated the great potential of using the Internet to build up virtual OKP enterprise with the capability of RPD. Hence, using the Internet to achieve RPD has become the most promising solution to meet the requirements of the next generation of OKP Systems.

The Internet and its relevant technologies (e.g. world wide web, communication, software tools and hardware) have greatly progressed in the past few years. The outstanding features of the Internet make it the best supporting platform for RPD in the distributed heterogeneous manufacturing environment (Cheng et al 2001). For example, the Internet platform has the capability to integrate diverse software tools for supporting applications. This facilitates the development of an IRPD systems with the integration of current technique for PD to support distributed team members to work together to design, manufacture and support products cooperatively and concurrently. The IRPD systems that are built based on the Internet platform can help manufacturing companies achieve flexibility, rapid responsiveness to the dynamic global market and the changing customer’s needs and rapidly produce and deliver products to the market.

On the other side, new technologies have been continually put forward and gradually updated to the Internet platform. They include Internet based CAx technology, Internet communication technology,
Internet based design for X, Internet based collaborative design, Internet based decision support, Internet based workflow management, Internet based CE, CAD/CAPP/CAM technology, Internet based virtual simulation technology, AI, knowledge bases, global optimisation technology and CSCW tools. An IRPD system that is developed by integrating these technologies using the Internet platform provides a solution for developing the next generation of OKP systems. The IRPD system will have great technological advantages over the traditional standalone environment or Internet based systems for supporting only a single stage of PD process. This IRPD system can consider the development process of OKP products as a whole and can thus achieve better performance.

2.3 Review of Internet based PD approaches

Internet based Rapid Product Development (IRPD) requires the integration of people, business processes and information technology across the PD life cycle for the purpose of RPD. A typical PD life cycle includes understanding customer requirements, product definition, product design, analyses and test/simulation, process planning, manufacture and delivering the product. This section will review the recent achievements in the areas of PD process and the emerging issues in terms of using them to improve the performance of IRPD, which includes Internet based collaborative design, Internet based design for X (DFX), Internet based decision support and concurrent engineering (CE).

2.3.1 Collaborative Product Design

The initial work on collaborative design appeared more than a decade ago, but it was until recently that industry, academia and government increasingly demonstrated the benefits of collaborative work (with the help of the Internet) for product design and manufacture. Collaborative design is the process of designing a product through concurrent cooperation among the engineers who are from different functional areas in a manufacturing company, e.g. design, process planning, manufacture, assembly, test, quality and purchasing as well as those from suppliers and customers (Sprow 1992). The collaborative design with geographically dispersed participants based on Internet communication allows participants to exchange data and thereby reduce the disadvantage of geographical dispersion. The objectives of such a collaborative design team might include optimising the mechanical function of the product, minimizing the production or assembly costs, and ensuring that the product can be easily and economically serviced and maintained (Harley 1998). This research area has been of tremendous interest to many researchers. As reported by Hartley (1998), the collaborative design can reap substantial benefits. As seen from the following data, it leads to a 75% reduction in design changes at Northrop, a 60% reduction in PD time at DEC, a 40% increase in customer satisfaction ratings at Xerox and a 60% reduction in scrap at General Motors.

As collaborative design can offer substantial benefits, it has been used in some areas. For example, collaborating via the Internet for design teams at Boeing-Rocketdyne (Carman 1998), research on using collaboration tools for manufacturers at NIST (Stieves and KnautilHa 1999), and efforts at understanding and supporting the design process at Stanford University (Cutkosky et al. 1996) all show encouraging results. With the development of other technologies (e.g. virtual reality and communication (Smith 1998)), the team members can achieve better visualization and communication, and thus better collaboration. These technologies enable collaborative design to take place and this will lead to more efficient collaborative design. In 1996, the NIST initiated a research and development program to help US industry speed the transition to 21st century manufacturing capabilities. This program is called National Advanced Manufacturing Testbed (NAMT). This Testbed contains a facility in which scientists and engineers from industry, academia, NIST and other government agencies work together to solve measurement and standards issues in information-based manufacturing. To achieve better collaborative design, several researchers have studied how designers work together in teams. For example, Minneman (1991) discussed social interaction among designers, which includes how design members in a design team with different opinions negotiate to reach common understanding. He also discussed how the integration could be used to improve design communication. Frankenberger and Birkhofer (1995) described engineering and psychological influences on designers, which include the influences on communication, e.g. leadership and
group organization and how a good team interaction leads to a good decision. To achieve collaborative viewing of mechanical part models on the Web, Kim et al. (2001a) developed a collaborative system called 3D-Syn. This system can support synchronous communication and manipulation of 3D part models on the WWW and can be used as an open platform for 3D collaboration to inter-link part suppliers and buyers. An object-oriented database is used for the part library that stores part information. As a result of the extensive research and development, there are some software tools in the commercial market for supporting collaborative design. The Collaborative Virtual Product Development (CVPD) software suite, for instance is developed to link global design teams and partners in a PD chain (http://www.findarticles.com/cf_0/m1068/23_55/69388478/p1/article.jhtml). Oracle (www.oracle.com) has entered the collaborative PD and product lifecycle management software market with the cooperation of several CAD vendors. PTC offers a version of its Windchill engineering management tool for the oracle product development exchange. (http://www.findarticles.com/cf_0/m0BLL/12_17/68645300/p1/article.jhtml).

Despite the success of the research as mentioned above, there are some research issues unsolved in collaborative design and manufacturing. For example, it is not clear how to implement the collaborative design and manufacturing in different manufacturing companies in terms of companies’ size, people, communication, various management styles and working environments. There are some cases reporting successful collaborative designs. These cases are generally concerned with engineers working at a single site and on a relatively uniform computer system platform. The question remains of how collaborative design can be best implemented in the integrated environment where there may be a wide variety of computing platforms.

2.3.2 Design for X (DFX) via the Internet

Design for X (DFX) has been recognized as one of the effective approaches to implement concurrent engineering for the goals of RPD (Xie et al. 2002b). Huang (1996) gave a clear definition of DFX, while D in DFX is interpreted as product design in the context of DFA (Design for Assembly), which means to design a product for the ease of its assembly (Boothroyd 1996). X in DFX stands for manufacturability, inspectability, recyclability, etc. Using the Internet to provide DFX services can be used to support the rapid and collaborative product development. DFX can be used to reduce the time and cost of redesign, assembly and manufacturing. As reported by (Bralla 1986), the DFX services are usually performed by using DFX guidelines, which include simple guidelines, advisory guidelines and quantitative guidelines. The simple guidelines are simple instructions, e.g. “avoid sloping surface”, “cutting rules” and “minimize the number of tools used in manufacturing” (Matousek 1963). The majority of guidelines fall into these so-called advisory guidelines. An advisory guideline usually includes a verbal description, and diagrams showing right and wrong design features. By following these guidelines, designers know what to avoid when they design products. Quantitative guidelines provide quantitative evaluation of how good or bad a design feature is in terms of certain criterion utilization (Carlsson et al.1994 and Boothroyd 1996). However, most guidelines developed in the past did not provide quantitative evaluation of the quality of a design in terms of certain criteria such as cost, time (Boothroyd 1996; Carlsson et al. 1994) and/or other aspects (Ishii et al. 1994). The latest developments tend to provide quantitative guidelines to improve the quality of design. Recent approaches on DFX delivery by using guidelines application have been carried out in various DFX areas such as design for manufacturing (DFM) (Lazar et al. 1992), design for retirement and recyclability (Ishii et al. 1994, Zhang et al. 1997a), design for environment (Stanley 1996) and design for ergonomics (Allada et al. 1992).

The Internet technology is becoming popular for the DFX analysis (Wagner et al. 1997, Huang et al. 1997). Recently, several Web-based systems have been set up, e.g. a Web-based design for assembly (Shi et al. 1995) and the Web-based failure mode and effects analysis (FMEA) (Huang et al.1997). The impact of Internet based DFX will be found throughout the overall product design and manufacturing process. For example, Internet based DFA techniques can be used to reduce the cost and time of assembly in a distributed manner by simplifying the product and process through such means as reducing the number of
parts, combining two or more parts into one, reducing or eliminating adjustments, simplifying assembly operations, designing for parts handling and presentation, selecting fasteners for ease of assembly, minimizing parts tangling, and ensuring that products are easy to test. Use of Internet based DFA to reduce the number of parts will reduce inventory, and so will reduce inventory management effort. As a result, it will support activities such as Just In Time (JIT) aimed at improving shop-floor performance.

Through the Internet based DFX systems, it is obvious that product developers should be able to access information that will help them improve the design of the part they are working with. These Internet based DFX systems can be called up to analyse the current state of their design, point out where the design is too complicated, and indicate possible areas of improvement. Companies using Internet based DFX techniques (e.g. DFA and DFM) have reported reducing the number of parts, the number of assembly tools, the number of assembly operations, the assembly space, the number of suppliers, and the assembly time by up to 85%. DFM helps prevent the unnecessarily smooth surface, the radius that is unnecessarily small, and the tolerances that are unnecessarily high. The DFA objective of reducing the number of parts may lead to highly integrated, complicated, and multi-functional parts. DFM aims to keep individual parts simple, because overly complicated parts can be hidden costs that are not initially apparent. At the early stage of the design, there may not be a lot of information to work with, but Internet based DFX systems that are based on DFX functionalities will make sure that whatever information exists can be made available to the product team, which is an important issue for RPD.

Although there are so many advantages of developing Internet-based DFX (IDFX) systems for RPD, the implementation of IDFX systems is never an easy task. Several problems have been raised for implementing DFX systems. Among these problems, an important question is whether there is a basic pattern for the development of these DFX tools (Olesen 1992). Another important issue is how to update traditional DFX systems to meet the requirements of the global RPD environment. These problems have become the major hurdles for developing and implementing the IDFX systems.

2.3.3 Internet-based Decision Support

Starting from the front-end of the PD process, the Internet-based approach is particularly suitable for some areas where decision supports are required, such as customer requirements management or market research, earlier design decision making and customer involvement in product design and manufacturing. The Internet based systems that can handle decision-making at various stages of PD process are vital for IRPD. The Internet technology enables us to develop virtually all types of decision support systems, e.g. PDM (product data management) systems, EDM (Electronic Document Management) systems and visualization and virtual tools (http://www.visualmining.com/). Owing to the multimedia capability (Weyrich et al. 1999, Ockerman et al. 1999) of the WWW and the success of remotely executing large programs (Su et al. 2001), Internet-based decision support systems are functionally better than standalone counterparts and compatible with heterogeneous environment. The literature and reports on various decision support systems/tools for stages of PD process (http://emt.doit.wisc.edu/decision.html and http://www.decision-support.net/).

An interesting work at the Philips advanced development centre is to use the Internet as a communication infrastructure to get users involved in the new product development process (Muller et al. 1996). Another related project is to use the WWW to analyse customer requirements for software development (Anton et al. 1996). As Internet technology is playing an increasing important role in marketing and sales as well as after-sale customer services, an Internet-based product and component catalogue has been widely used. Active Catalogue (Will 1996) is a project aiming to develop WWW-based component catalogues including models to enable ‘try before you buy’ simulation analysis during a product purchasing process, which can be regarded as a decision support system for rapid product sell. HKCAINS (Hong Kong Accessory Information Network System) is an industry-based project aiming at developing a WWW-based network for the Hong Kong apparel industry to mutually communicate and seeking advice between each other as well as to assess the economic viability of establishing this network (HKTAIGA 1996). Wong et al. (1996) proposed methodologies for a rapid and accurate response to
request-for-quotation and demonstrated the method on the Internet. The key issues that should be considered when developing Internet based decision support systems are identified as follows: (a) distributed network of computers; (b) sharing of data; (c) sharing of tools; (d) tracking of data (Cheng et al. 2001). The Internet can also deliver functionalities of decision support through developing Internet based virtual systems. One potentially significant project is ‘Intelligent Manuals’ (Pham 1998). The objective of this project is to supply the electronic information necessary to support the continuous use and maintenance of a product from its delivery to its disposal. Kim et al. (1998) proposed a WWW-based architecture for collaborative design in mixed platforms and dispersed geography environments. They used the open data standards to allow users on a wide variety of platforms to access and visualize product information. Kalyanaprasad et al. (1997) proposed a system using the Internet to support the generation of group technology codes for mechanical parts. IAMS (intelligent assembly modelling and simulation) aims to facilitate assimilability checking in a virtual, simulated environment in order to avoid expensive and time-consuming physical mock-ups. It is in fact one project within a broader effort of developing collaborative open design system (IAMS 1998) and acts as decision supporting. Szykman, et al. (1999) developed "Design Repositories" to support the integrated and concurrent design process. They also pointed out that the "Design Repositories" was a key issue to provide a comprehensive product knowledge representation during the design stage. It is obvious that developing products without sufficient support information in a broad set of disciplines would result in longer PD cycles, higher development cost and quality problems.

However, for these approaches, there existed some limitations to their applications. The first is the requirement for a formal WWW-based distributed database structure with suitable product data models for a manufacturing company, i.e. a decision must be made on what information should be shared and how to represent and record the information from a product development and the relevant tool/mould making processes. In practice, the company production manager usually makes his/her decisions about what information is needed, when it is needed and how it will be used depending on the context of the current problem (Boynton 1993). Also, the data structure of the product information management system may vary with the structure and culture of the company. Considering these questions in advance, collecting all the critical pieces of information for the decision, and finally integrating the information and decisions into a product model is usually a hard job (Gruber et al. 1996).

### 2.3.4 Internet based Manufacturing Scheduling, Planning and Control

Manufacturing scheduling, planning and control is a difficult problem for developing OKP products, particularly when it takes place in an open and distributed environment. In the manufacturing process, things rarely go as expected and dynamic changes are often needed. To save product manufacturing time and cost, Internet based manufacturing scheduling, planning and control has become an important research field in recent years. As the manufacturing environment in a company often involves a variety of machining systems, monitoring facilities, control equipment, and information resources, an efficient and easy-to-use client-server manufacturing scheduling, planning and control system is vital for global enterprises. For example, it facilitates scheduling job tasks among different machining systems, and provides fast data or information exchange between subsystems and/or terminals and rapid changes from the network to a CNC machine. Hence, various manufacturing systems/facilities distributed at different companies can be organically organized and shared under their agreements.

In order to develop systems for distributed scheduling, process planning and control, tremendous efforts are made by many researchers. As reviewed by Shen et al (1999), there were 30 projects that were being carried out all over the world. For example, Shaw proposed using agents in manufacturing scheduling and factory control. He suggested that a manufacturing cell could subcontract work to other cells through a bidding mechanism (Shaw and Whinston 1983, Shaw 1988). YAMS (Yet Another Manufacturing System) (Parunak 1987) was also one of the earliest agent-based manufacturing systems where in each factory the factory component is represented as an agent. Each agent has a collection of plans, representing its capabilities. The Contact Net is used for inter-agent negotiation. Zhang et al (1997b) proposed an integrated approach for process planning and production scheduling by building up an integrated manufacturing

There are also approaches to achieve production planning and control by using Internet based supporting systems. Zhou et al. (1999), for instance, proposed an information management system for production planning and control in a virtual enterprise under a distributed environment. Ockerman et al. (1999) used multimedia technology in their project called Factory Automation Support Technology (FAST) to improve worker performance throughout the factory. The IPPI (integrated product processing initiative) project is another major effort aiming to develop and validate a prototype process planning system that will utilize form feature product models defined in STEP format and is capable of generating intermediate product models representing the state of the product prior to and subsequent to each manufacturing operation. The goal of the IP3S (integrated process planning/ production scheduling) project is to dynamically convert standards-based product specifications into process plans and schedules that best accommodate current shop load, the status and allocation of machines, fixtures and tools, and raw material availability, while minimizing production costs, lead times and inventories, and maximizing due date performance.

Typical research issues raised from recent researches in this area can be sorted into following three categories. First, there is not a common model for developing Internet based scheduling, production planning and manufacturing systems for various applications. Second, issues in agent technology for the dynamic manufacturing scheduling, planning and control, e.g. integration of planning and scheduling (Maturana et al. 1997). Third, research issues in scheduling and planning algorithms, e.g. dynamic executing of remotely large programs and algorithms (Su et al. 2000).

2.3.5 Internet based Concurrent Engineering

CE was defined by Cleetus (1992) as “a systematic approach to integrated PD that emphasizes response to customer expectations and embodies team values of cooperation, trust and sharing in such a manner that decision making proceeds with large number of parallel working by all life-cycle perspectives, synchronized by cooperatively brief exchanges to produce consensus”. CE has been used in a wide variety of applications for the purpose of improving the performance of RPD, e.g. reducing the PD cycle time and cutting down costs (Aldanondo et al. 1997, Young et al. 1992). It has been found from contemporary research in the fields of CE that significant benefits can be achieved if suppliers are involved in the new product development process as early as possible. However, recent investigation in manufacturing industries has also revealed that this approach is not widely practised in industries and its implementation has been a great challenge to researchers and practitioners.

In the distributed PD environment, CE teams would include various functions as disparate as design, planning, manufacturing, assembly, test, quality and purchasing as well as including suppliers and customers (Sprow 1992). The overall goals of CE include shortening time to market, reducing the costs of the total product life cycle, and increasing quality. The objectives of the CE might also include the optimisation of the mechanical function of the product, minimizing the production costs, reducing the PD time and ensuring the product can be serviced and maintained both easily and economically.
Chapter 2 The State-of-the-art Literature Review and A Proposed Reference System

Structure

To support CE in the distributed environment, the international research community has carried out extensive research in order to establish ways of supporting engineering activities. However, the term of Internet based distributed CE contains new concepts, as the team participants are globally distributed and use a wide variety of computer systems. A few research projects have been mentioned in the literature on how the Internet based distributed CE can be best implemented, particularly for a globally distributed CE team that may be using a wide range of computer systems. The requirements for building an environment to support the Internet based distributed CE are identified as follows (Kang et al. 1997): 1) The environment should use open standards, since such open standards allow participation by a wide variety of parties, including suppliers and customers. 2) The environment should allow natural communication between team members. 3) A Product Database Management System (PDMS) should be used to manage the storage and retrieval of the large amount of information for a reasonable sized company. Such a PDMS should be allowed to integrate with other functions within the company. 4) The environment should support interactive 3D graphical representations of parts and products, these parts and products should be available for users to manipulate. 5) The environment should have effective data conversion engines to support data transfer between data stored in PDMS, the standards used for data transfer, the visualization standards and data standards of various CAD, CAPP and CAM systems.

These requirements pose some challenging issues to researchers in this area. As identified by Christiansen et al. (1996), CE can be realized through three different non-exclusive approaches: 1) Organizational approach, where engineers from different departments are organized in PD teams; 2) Non-IT approach, where specification activities are supported with, for instance, technical manuals describing a group of similar products (or components) and their manufacturing specifications based on a group technological analysis of products or components; and 3) IT approach, for instance, using features and product modelling to structure knowledge and information for specifying products in the different phases of the product life cycle. Among these approaches IT approach is the most promising approach to achieve Internet based CE. However, this research is still at its very early stage. Further research in terms of the supporting technologies and the application of CE in the whole PD process are required.

2.4 Review of IRPD Approaches

Besides the research topics reviewed above, great efforts are also made in other fields of PD, e.g. workflow managementsuch as WebFlow (WebFlow Corporation 1997), PrISMS (1996), OzWeb (Kaiser, Dossick and Yang 1997), resource planning, marketing and supply chain management (Huang et al. 2001b), etc. These extensive research efforts that have been made in different areas of PD raise the possibility that IRPD systems can be developed within the integration of these technologies. To improve the agility and responsiveness of manufacturing enterprises and to enhance the ability to rapidly combine the strengths of the manufacturers and suppliers, some researchers have also made their contributions in this research area, although the functionalities of the Internet based systems are limited. Some of the research approaches are briefly reviewed in the following.

1. Collaborative approach

To shorten a PD cycle, the interdisciplinary team that works collaboratively is one of the features of RPD. Various attempts have been made to develop Internet based collaborative systems to support rapid product design and manufacturing using collaborative technologies. They include the research and development of collaborative technologies (Steves et al. 1999), CSCW tools (Bullinger et al. 1996), collaborative engineering database, engineering information framework (Szykman et al. 2000), engineering knowledge structure (Xue et al. 1999) and the development of Internet and information based collaborative systems (Cheng et al. 2000). These collaborative approaches aim to achieve better cooperation among distributed team members and are of great importance for RPD. There are also tools and methods proposed for collaboration over the Internet. Some of the typical works are reviewed as follows. Chui and Wright (1999) developed an Internet-based multi-media educational tool for the design of simple mechanical parts. Bather (2000) proposed a multimedia communication framework for the selection of collaborative partners.
in global manufacturing. The proposed framework was generated from the CIM-OSA (Open System Architecture for CIM) approach and modelling with the aim of simplifying the enterprise’s collaboration. Reed and Afjeh (1998) have used web-based interactive engineering tool for engineering simulation and teaching and learning purposes. One of the early web applications is to provide rapid prototyping services on the Internet (Bailey 1995, Wright et al. 1997). Smith and Wright (1996) collected a number of WWW-based design and manufacturing services for their cyber cut experiment. Roy et al. (1997b) have presented experimental workbenches for WWW-based design to production, including activities, e.g. conceptual and detail product design, process planning, design for manufacture, NC programming and rapid prototyping.

2. AI & knowledge Approach

The increasing complexity of products and processes require earlier decision-making, the tools that can help decision making at the early stages of essential for RPD. Hence, knowledge and knowledge modelling on design, process planning, quality, etc. and AI for supporting decision-making has become an important research field of the RPD. In recent years, works have been carried out to develop Internet based intelligent systems to support RPD. For example, Pan et al. (1997) integrated AI and knowledge technologies to greatly improve the agility of product design and manufacturing. The integration of these technologies with Internet technologies is the enabling technology for distributed manufacturing companies to achieve short PD cycle times and to respond quickly to sudden market opportunities. Cheng et al. (1997) and Pan et al. (1999) proposed a Java and AI-based system for the implementation of design agility and manufacturing responsiveness. Ambrosio et al. (1997) proposed a World Wide Web approach for capturing and deploying the preferential knowledge required to resolve design conflicts. Xue et al. (1999) presented methods for modelling knowledge base and database for intelligent concurrent design. A system that combines knowledge based reasoning and optimisation was also introduced to automatically generate the aspect models and identify the optimal design using optimisation. Xie and Tu (2000) proposed a WWW based integrated PD platform for intelligent and concurrent sheet metal design and manufacturing. The platform integrated knowledge base, AI and Internet technologies for sheet metal design and manufacturing. Cheng et al. (2000) presented a novel approach to implementing agile design and manufacturing concepts by using Internet-based technology. The underlying philosophy of the approach is to use Web-based design and manufacturing support systems as smart tools from which design and manufacturing customers can rapidly and responsively access the system’s built in design and manufacturing expertise.

3. Integrated and concurrent approach

As collaboration and partnership become very important in modern manufacturing companies, CE (Young et al. 1992, Kim et al. 1998) and integration methodologies (Chang 2000) are recognized as successful ways to support RPD. This area is also a strongly emerging research field. A few research projects addressed the importance of the integration and concurrent approach in the PD process. For example, Xie et al. (2000) proposed a WWW based integrated PD platform for intelligent and concurrent sheet metal design and manufacturing. This platform integrates various software tools for sheet metal design, unfolding, planning, cost optimisation, manufacturing and marketing. Zhang et al. (1997b) proposed an integrated manufacturing environment for the integration of process planning and production scheduling. Tu et al. (2000b) presented a virtual PD platform for RPD with the integration of specific models for the involvement of customers. Chui et al. (1999) presented a WWW computer integrated manufacturing environment for rapid prototyping and education. Reed and Afjeh (1998) used a WWW-based interactive engineering tool for engineering simulation, teaching and learning purposes. Researches have also focused on the ways of supporting integration and concurrent approach, for example, to integrate the available information is a key step for the different partners in a PD cycle to effectively share the information (Dong et al. 1998). Chen et al. (1998) developed an integrated graphical user interface for CE design of mechanical parts, and several typical models were suggested in their paper, i.e. a WWW-based on-line user’s guide, a part library, a design guideline checklist, a part modeller linked to a CAD system (Pro/Engineer) and a knowledge based design critique system. Su et al. (2001) proposed an Internet-based system for
geographically dispersed teams to collaborate over the Internet for the purpose of integration in design and manufacture. A CGI (common gateway interface) -based multi-user method has been developed to remotely execute large-size software systems via the Internet.

Great research efforts have also been made to develop an integrated information-sharing platform for the purpose of supporting integrated and concurrent PD in a computer network environment. Some prototypes have been developed though they are still far from commercialised, e.g. Dong et al. (1998), Xue et al. (1999), Cutkosky et al. (1993), Boynton (1993) and Gruber et al. (1996). Chang (2000) presented an empirical study of the implementation and integration of information systems for production management in manufacturing. Shackleford and Proctor (1998) developed the Java-based tools for development and diagnosis of a real-time control system. Furthermore new approaches have been proposed to implement WWW-based integrated systems for product design and manufacturing. They have been used in different areas, e.g. agile manufacturing (Cheng et al. 2000), sheet metal concurrent design and manufacturing (Xie 2001), collaborative design of 3D mechanical parts (Kim et al. 2001a). With the support of the European Union, an international team has worked on a collaborative research project called a global engineering network which aims to provide a global collaborative design platform across various EEC countries (Gausemeier 1996).

2.5 Review of Implementing Technologies for Developing IRPD Systems

Although the research for Internet based PD systems as reviewed above has been started for several years, still no united way has been found to develop IRPD systems. The implementation of IRPD systems covers broadly current technologies. Internet applications can be implemented in many programming languages, tools and environments. The most popular technologies that can be used to develop the IRPD systems include, DCOM (Distributed Component Object Model), CGI, STEP, SDAI (STEP Data Access Interface), CORBA (common object request broker architecture) and ActiveX technology and JAVA and agent technology. This section gives a review of how these technologies can be used for developing IRPD systems.

1) ActiveX and DCOM technology

a. ActiveX

The Microsoft ActiveX technology allows programmers to assemble reusable software components into sophisticated applications and services in an Intranet/Internet environment easily with minimum efforts (Swank et al. 1997). This technology has been used in the development of various Internet based research projects (Xie et al. 2001a, Huang et al. 2001a). One of the features is that the software components called ActiveX components can be reused by different software platforms. In fact, ActiveX components can be created using Microsoft Visual Studio toolkits, e.g. Visual Basic, Visual InterDev, Visual Java++ and Visual C++. Several types of ActiveX components are available to develop in both applications at server and client sides. ActiveX components can be used to develop applications that integrate tightly with the other elements of the Internet or intranet site. Application clients can be compiled into ActiveX controls that can then be embedded into HTML web pages, or into ActiveX documents that are attached to HTML web pages but are usually executed in a separate container. The downloading, installation and execution processes of both ActiveX controls and documents are similar. When a user accesses a URL with an ActiveX component, it is downloaded from the server to, and then registered on the client machine along with the HTML page that uses script to invoke the ActiveX component.

Huang et al. (2001a) detailed the functionalities of the ActiveX in documents and controls. ActiveX controls are usually embedded into HTML web pages. ActiveX documents are non-HTML documents that can be viewed and edited in a web browser. The HTML page is replaced by the ActiveX document, and the ActiveX document executes in the web browser as its application container. ActiveX documents also offer more complex client-side processing than HTML pages with ActiveX controls. For example, ActiveX
documents may have all the elements as stand-alone systems, e.g. pull-down menus. ActiveX controls do not have such capability as pull-down menus. An ActiveX code component is an object or objects exposed by an application that can be controlled programmatically by other applications. ActiveX code components can be used to add functionality to an HTML page on the client-side just as with ActiveX controls. However, it is more often used to deploy application servers as ActiveX code components. Server-side ActiveX code components can be used to customize the creation and return of an HTML page, just like CGI programs. In addition, they can be used to manage a database connection, to execute a standalone algorithm and marshalling queries received and results returned.

b. DCOM

The DCOM architecture is widely used across a broader scale of multi-user applications. It can be used in most of the application cases in developing Internet based PD systems. DCOM is an ideal technology for multi-tier applications because it enables ActiveX components to work across networks, enabling developers to easily build systems that span computer boundaries. The DCOM has three unique strengths that make it as an important technology for IRPD, i.e. 1) DCOM is based on the most widely-used component technology; 2) DCOM is simply "COM with a longer wire"—a low-level extension of the Component Object Model, which is the core object technology within Microsoft ActiveX. Major development tools (e.g. Microsoft, Borland, Powersoft/Sybase, Symanetc, ORACLE, IBM, and Micro Focus) and the applications they produce automatically support DCOM, providing the broadest possible industry support. Additionally, over 1,000 existing commercial software components that will work with DCOM are already available for use by developers. 3) DCOM is an open technology that runs on multiple platforms. Microsoft is openly licensing DCOM technology to other software companies to run on all of the major operating systems, including multiple implementations of UNIX-based systems. As recognized by many developers, it is the best networking technology to extend component applications across the Internet.

The combination of these three advantages—the largest installed based native support for Internet protocols, and open support for multiple platforms—means that businesses can gain the benefits of a modern component application architecture without having to replace investments in existing systems, staff, or infrastructure. Developers add components together without having to worry about network programming, system compatibility, or integration of components built from different languages. This can lower the cost and complexity of building distributed applications from components. DCOM leverages the investments companies have already made in ActiveX by providing the following benefits: 1) Multi-platform support—DCOM is designed to run on Windows 95, Windows NT, Macintosh, UNIX, and legacy operating systems, providing companies with the basis for a common application infrastructure across their entire IT environment which can lower integration costs and reduce integration complexity; 2) Evolutionary technology—In addition to Java support, DCOM enables components written in other languages, including C, COBOL, Basic, and Pascal, to communicate over the Internet, providing a growth path for existing applications to support Web technology; 3) Common components for the browser and Web server—Since ActiveX components can be embedded into browser-based applications; DCOM enables a rich application infrastructure for distributed Internet applications using the latest browser technology; 4) Security—DCOM integrates Internet certificate-based security with rich Windows NT-based security, combining the best of both worlds; 5) Standards-based—Microsoft is working with Internet standards bodies, including the IETF and the W3C, to offer DCOM to the Internet community as an open technology. DCOM is based on the Open Group DCE RPC, an open and widely deployed communications technology. The DCOM wire protocol extensions have been submitted as an Internet draft and are available at http://www.dce.luth.se/doc/id/draft-brown-dcom-v1-spec-00.txt.

2) Java technology

Java technology has a broad application in PD processes (Cheng et al. 1997, Shackelford and Proctor 1998, and Pan et al 1999). Java was originally proposed by Sun Microsystems (java.sun.com). This software development platform can be used to develop Internet based systems based on the Internet platform. It is a
portable, object-oriented, distributed and multitthreaded programming platform. All Java programs can run in Java Platform that has two components: the Java virtual machine (Java VM) and the Java application-programming interface (Java API). The Java VM makes JAVA application programs that can run in any operating systems, which is important for the integration of the different manufacturing systems that run in a heterogeneous environment. Java API provides a variety of functions for users to develop different applications, e.g. Java SDAI provides tools for accessing STEP based databases, Client/ server tools, etc. Figure 2.1 shows an integrated environment for Internet programming by using JAVA script in both server and client sides. This environment has connections with data resources from networked databases and provides efficient server-side programming. Java applets depend on web browsers for their installation (downloading) and execution in the client machine.

In this integrated environment, the JavaBeans, which are components built in Java, can be used. This JavaBeans specification describes new component architecture for Java to facilitate component code development and reuse. As the JavaBeans component model is based on Java classes, this model can be well extended and reused, e.g. simply adds a few rules and these rules can be used with the class. Beans can be deployed as servers and clients. A bean is a reusable component that can be used to create applets, applications or even HTML pages. The same bean should be able to play in any of these containers. The emergence of the JavaBeans component model is expected to further simplify the web application development in the Java environment.

3) CORBA and Agent technology

a. CORBA

CORBA, developed by the Object Management Group (OMG), has become a promising approach to the development and implementation of distributed systems (Harmon and Morriscey 1996). It has many distinguishing advantages over other approaches (e.g. SQL, email based groupware and transaction process monitoring approaches). It can avoid direct links between applications via defining a broker that acts as an object bus, and can also encapsulate existing legacy applications. These characteristics are extremely suitable for WWW based systems development (i.e. virtual enterprise). Another important point of the CORBA technology is that it can be used for network communication between different computer system platforms. According to the CORBA standard, all software systems written in different programming languages on different platforms can exchange information via the interfaces defined by the CORBA IDL (interface definition language) and ORBs (Object Request Brokers).

The OMG’s CORBA technology can be well used in Internet as the communication mechanism. The ORB interoperability allows communication between independent implementations of the CORBA standards. The OMG also defines a specialization of general inter-ORB protocol called Internet inter-ORB
(IIOP). CORBA has its inherent advantages in heterogeneous distributed object computing environments (Vinoski 1997) as CORBA specification has defined six language mappings from OMG IDL to the most popular programming languages and available CORBA products can support almost every popular operating system. The new generation of CORBA as mentioned in http://www.cs.wustl.edu/~schmidt/corba-research-overview.html will optimise its performance and improve its communication speed and safety. The integration of CORBA, STEP and agent technology has shown a promising future to meet various needs of developing advanced manufacturing systems. A more detailed technical introduction of CORBA technology can be referred to http://www.omg.com.

As mentioned in section 2, the characteristic feature of the RPD environment is heterogeneous in its computing and communication environment. This not only includes computing platforms, operating systems and network protocols, but the applications utilized in different departments, i.e. CAX, DFx, PDM, etc. In such an environment, the CORBA technology becomes very important. In recent years, numerous approaches using CORBA technology in manufacturing were reported (Coulson 1998). For example, Whiteside et al. (1997) reported a CORBA based manufacturing environment for Sandia’s Agile Manufacturing Test-bed (SAMT). This environment uses CORBA technology to support information integration, sharing and cooperation among distributed manufacturing cells. An Integrated Simulation-Based Design and Manufacturing Environment using Intranet and CORBA was reported by Edwin et al. (http://www.ccad.uiowa.edu/~infoint/nsf/papers/dls/DDSPaper.html). This environment enables the achievement of concurrent engineering goals by using CORBA standard as a network communication standard. It gives the members of an enterprise-wide product development team a convenient, uniform interface to the global product data from different platforms. Howard et al. (1996) used CORBA and STEP to develop the next generation standards and measurements needed for information-based manufacturing. The systems that developed based on this standard can be used to support strategic partnerships among diverse and geographically dispersed companies. They are particularly effective at carrying out needed manufacturing operations and achieve integration by efficiently managing information flow between different manufacturing systems.

b. Agent technology

In recent years, agent has become a popular technology for improving PD process. Although the term ‘agent’ has been used widely, it has no a unified meaning, definition or structure (Lei et al. 1998). It can be a hardware or software component (Shoham 1993), or a combination medium between human users and software tools (Genesereth and Nilsson 1987, Khedro 1996). However, many researchers in different fields have implied or clearly suggested that an agent possesses certain fundamental characteristics and capabilities (Crowston and Malone 1988, Shoham 1993). First, an agent possesses and maintains certain information. Second, it is able to interact with its environment and extract knowledge from it. Third, the agent is able to communicate with other agents for information and knowledge exchange. Fourth, it is able to process information, messages, and events and make decisions autonomously. Last, the structure and contents of an agent can be changed or defined by users through using a software platform. In the IRPD environment, an agent is defined as a software program that has the fundamental characteristics and capabilities as mentioned earlier. These characteristics and capabilities of the agent technology enable the multi-tasks adjustment and control, which act as “coordinators” for the scheduling, planning and manufacturing in a distributed environment. As discussed in section 3.4, agent technology has been widely used in diverse PD processes, e.g. distributed manufacturing, scheduling, planning and control (Gyires and Mathuswamy 1996, Pan and Tenenbaum 1997, Sikora and Shaw 1997).

4) STEP and SDAI

As a wide variety of software systems are applied in product development or production, e.g. different kinds of CAD, CAE, CAPP, CAM, scheduling, and MRP systems, it has formulated a PD computer system environment. In this heterogeneous computer based system environment, the STEP is crucial for data communication and exchange between these systems. STEP (ISO 10303) is an emerging
international product data standard that uses a high level, feature-based and object-oriented approach to define products. It is able to provide a complete, unambiguous, computer interpretable definition of the physical and functional characteristics of each unit of a product through its life cycle. STEP is organised as a series of parts according to description methods, application protocols, implementation methods, and conformance testing. STEP uses a formal specification language EXPRESS (Schenck et al. 1994) to specify the product definition information. The functionaries of STEP have been addressed by ISO STEP handbook. Although STEP is still in its developing stage (Mannisto et al. 1998), it has been used in various industrial fields. Various STEP-based application protocol (AP) tools have been developed and used in PD processes, e.g. modelling generic product structures (Mannisto et al. 1998), data migrating and translation (Mangesh et al. 2000a), information modelling (Lee 1997), feature extraction from STEP geometry for agile manufacturing (Mangesh et al. 2000b), building an integrated environment (Yang et al. 1999) and Internet based data exchange framework (Zhang et al. 2000).

SDAI is a tool that is provided by the STEP community interfaces for data sharing and exchange. Usually there are three kinds of data sharing methods: 1) Exchange file; 2) SDAI; 3) Database management system. Among these, implementation problems make the third option difficult. Sharing by means of files is just a static data sharing. High level of standardisation is on the way for data sharing using SDAI, which is a dynamically way of sharing data. This feature is very important for dynamic information exchanges. SDAI standardizes the runtime interface to STEP data, which enables application softwares to develop functions to dynamic operate STEP data. Commercial languages (e.g. C, C++, FORTRAN and Java) provide application packages for SDAI. For example, SDAI/Java, as a binding language, has been used in various product developments. Portability of Java enables SDAI to be easily ported to any node in a heterogeneous enterprise network. Hence, SDAI has become an important tool for developing manufacturing systems that use STEP as a standard. Some case studies are available in (http://mega.ist.utl.pt/~ie-sdai/docs/).

2.6 Research issues and Future Trends of IRPD

Despite the great contributions that have been made from numerous research and development projects to develop Internet based systems to support various PD processes as reviewed in section 2.5, there are still some major issues that hamper the developing of IRPD systems, especially when it comes to the integration of those technologies. These problems include the requirement for a formal Internet-based distributed data environment (Urban et al.1999), knowledge and AI sharing and support, global optimisation algorithms (Tu et al. 2001) and Internet communication issues (Huang et al. 2001b). These issues, which are briefly summarised in the following, need to be solved before developing IRPD systems.

2.6.1 Implementing Issues

The Internet technology tends to be used by most of the global manufacturing companies. Just as reviewed in section 2.5, some newly developed Internet communication technologies have been used to develop various PD systems. However, it is still not clear how these developed technologies and systems can be effectively implemented to support the IRPD processes. It is also not clear what kind of things in PD can be extended and added to the Internet based systems. Principally, all the systems that are involved in the PD process can be implemented by using Internet technology. In addition to these implementation technical problems, the integration of these technologies and their applications is another trouble to implement the IRPD systems to manufacturing companies. People in the field of product design and manufacturing, for instance, have limited knowledge of Internet technologies, whereas the IT people normally do not have enough knowledge of product design and manufacturing.

So far, to my knowledge, no literature and research report has been found to review and summarise the complete implementation issues, e.g. how to develop Internet based systems, and what kind of systems are suitable for upgrading to the Internet, how to integrate design and manufacturing through the Internet and how to select Internet based systems for different companies, etc. Usually, the following questions
should be asked before considering research or deploying Internet based product design and manufacture systems. These questions are:

1. What product design and manufacture problems are most suitable for web applications?
2. How advantageous are the web applications over standalone and usual client-server applications?
3. How are sound web applications developed, implemented, deployed and applied in the field of product design and manufacture?

For developing IRPD systems, there are a number of issues that manufacturing companies have to face besides the issues above. The first issue is what is the proper structure (e.g. network structure, communication protocol) for the IRPD systems under a heterogeneous working environment. The second issue is how to select a programming environment that can easily work within the Internet environment and supported by recent Internet technologies. The third issue is how to make use of existing technologies that described in section 5 to resolve IRPD issues. The fourth issue is what kind of data environment should the IRPD systems be based on. The last issue is how to make use of existing systems (e.g. how to upgrade existed standalone systems to the Internet, connect existed product data management systems with the Internet). Other issues, e.g. securities, ownership, etc. should also be considered before even starting the development of the Internet based approach.

2.6.2 Systems Integration Methodologies

As discussed in section 2, IRPD requires the integration of people, business processes and information technology across the PD value cycle. This requires interfacing and integration across organisational functions as well as suppliers and customers. For the development of manufacturing technologies and the emergence of concepts like computer integrated manufacturing (CIM), there is a need for extensive cooperation between different engineering activities. More recently, the emergence of new manufacturing paradigms has taken the concept of cooperation to a higher degree. Product related information like design, manufacturing, utilization, maintenance and disposal is not only shared among various departments within a company, but is shared among various "partner" companies around the world, e.g. sharing of new geometric algorithms (Wagner et al. 1997). This requires a new "generation" of systems integration, which not only includes information sharing, but also the integration of technologies for design, planning, simulation and manufacturing. As efficient and effective information exchanging among different systems and different people in different PD teams becomes vital for IRPD, intensive investigation into how best to integrate various Internet based systems that are used at different stages of PD process through the Internet is required. Although the Internet has made a variety of types of information accessible to multiple users simultaneously, it still does not offer any simple way to use standalone applications (Tomarchio et al. 2001), particularly to those computationally and graphically intensive applications often required at various levels in a manufacturing environment. There are many factors that affect the integration of Internet-based standalone support systems for design and manufacturing. The obstacles to their applications are also from the heterogeneous manufacturing environments existing in a company or among enterprises. Furthermore, it is difficult for any single software supplier to have the necessary expertise to support a full suite of software modules which are used in an enterprise to facilitate its various functional activities. Hence, new system integration techniques are the key issue for developing IRPD systems.

In the early 1980s, manufacturers such as Xerox, Boeing, and GE and the vendors Digital and Computervision realized that true integration at the workgroup level between multiple CAD and CAE applications was an extremely difficult problem to solve because of incompatible geometric representations and proprietary database and file structures. In recent years, despite great efforts having been made to enable system integration, e.g. integration platform by Tu et al. (2001), smart drawing under networking environment by Andy et al. the PACT project by Cutkosky et al. (1993), etc., further research and development are still needed in future since the issues in systems integration have more and more influence
on the performance of RPD. The research issues for system integration mainly include three areas (Fowler et al. 1995), which are: 1) technology development, 2) standards development, and 3) tested and technology transfer. The technology development is to develop the necessary technologies for systems integration. The following research issues have been identified: 1) life-time data model for systems integration (Tu et al. 2001), 2) determining attributes of a complete and consistent model for PD process, 3) test platform for the integration mechanisms for systems integration. Secondly for the standards development, two research issues need to be addressed: 1) reengineer critical standards development processes, 2) improving standard & system interoperability. Finally for the tested and technology transfer, three key problems need to be studied: 1) how to establish computer support systems for an electronic network, on-line databases, and user-friendly search techniques, 2) how to achieve the transfer of technology over the Internet, 3) the development of a supporting environment and communication testbed to perform tests. These issues and problems will continually be the hot-spot research topics in the world.

2.6.3 CSCW and Interfacing Techniques

As IRPD requires the extensive cooperation of various PD systems over the network environment, the Computer Supported Cooperative Work (CSCW) and interfacing techniques are becoming the pinpoints for developing IRPD systems. They directly affect the accuracy and efficiency of the data management through a whole PD cycle. However, so far the development of CSCW technologies has not been so successful (Lubich 1995). The main problems include: 1) the technology is not available to all in the relevant community, 2) the current technology does not support multi-interaction among various systems, 3) the technology is not compatible with normal work, and 4) a set of rules to govern the CSCW work to meet with social and organizational norms is still missing. A further problem with the CWCS tools is the assumption of fixed models of users or organizational cultures, which do not meet the actual requirements of malleability and linkability (Simone et al. 1995).

The interfaces in the IRPD systems should have the following capabilities: 1) workbench (common users interface to multiple PD applications), 2) process modelling and workflow, 3) application encapsulation and invocation, and 4) application data translation and transfer and multi-vender networking and platform support. The commonly used data exchange standards, e.g. CAD-NT (CAD-Normteile), CGI (Computer Graphics Interface), CGM (Computer Graphics Metafile), SET (Standard data Exchange Transport), PDES (Product Data Exchange Specification), STEP (Standard for Exchange of Product Data), and IGES (Initial Graphic Exchange Specification), do not have the functionalities to directly support interfacing via variant application software packages. In recent years, tremendous efforts have been made to solve this problem. For example, Lau and Jiang (1998) reviewed the state of the art data modelling and exchange in enterprise integration and found that companies mainly used neutral format file to exchange data between different systems. Tu and Xie (2001) presented an integrated product model which can be used to record all the necessary data through a whole PD cycle into a common object to achieve data sharing across diverse engineering applications. This product model or the common object can be manipulated directly by all kinds of computer aided engineering software systems. As the representation of non-geometric information by using STEP is still at a preliminary stage (Wang et al. 1995) and industry’s reliance on non-geometric information and knowledge-based design and manufacturing increases, Tainfield (2001) presented an advanced life-cycle model for complex product development. The life-cycle model called SICODEL was designed to organically incorporate the ideas from various existing paradigms, such as CE, life-cycle engineering, DFX, production planning and control, virtual prototype, intra-inter enterprise information, rapid prototype and evolutionary life-cycle models of software engineering into a whole integrated information system concept. Dorador and Young (1999) explored the definition of the structure of information models for supporting the related processes of design for assembly and assembly process planning to support the intensive sharing of information required in CE throughout the life cycle of the product. Lee et al (1998) reported their experiences on developing information models for a variety of manufacturing domains, such as plant layout, process planning, discrete-event simulation (Ellis et al. 1998), and apparel pattern making for the SIMA (Support Initiative for Multimedia Applications) and NAMR
Chapter 2 The State-of-the-art Literature Review and A Proposed Reference System

Structure

(National Association of Manufacturers and Representatives) Programs. Three data-modelling methodologies were adopted in their research, i.e. the entity-relationship (ER), the functional modelling and the object-oriented (O-O). Three major information modelling languages such as IDEF1x (the Integrated Computer Aided Manufacturing (ICAM) Definition Language 1 Extended), EXPRESS and UML (the Unified Modelling Language) were used for building the information framework. They also suggested that effective information sharing and exchange between computer systems throughout a product life cycle has been a critical issue. To achieve data communication and exchange between different systems under the Internet environment, Zhang et al. (2000) presented an Internet based STEP data exchange framework for virtual enterprises. An Internet based services translator for Non-STEP and STEP data communication and exchange between heterogeneous computer systems was proposed. Urban et al. (1999) presented an integrated product data environment (IPDE). The IPDE goes beyond the functionality of current product data management tools in which it deals with product data at a semantic level of using the STEP.

Despite the success of the above research, the research of CSCW and interfacing techniques for the purpose of RPD is still in its developing stage. Some issues have been identified as main hurdles, e.g. using STEP representation of non-geometric information is at a far more preliminary stage (Wang et al. 1995), the combination of the Internet modelling methodology (e.g. HTML and CORBA) with STEP is not yet mature (Hardwick et al. 1996), and the Internet based data translators between various data formats are still at the early research stage (Zhang et al. 2000). These would be the major research areas for achieving the RPD.

2.6.4 Global Cost/Lead Time Optimisation

In order to make the PD process quicker and hence to produce cheaper and better products, the optimisation of the key parameters of the PD process is necessary and it has become very important for PD. There are some optimisation algorithms proposed for improving the products' competitiveness, e.g. Multidisciplinary Design Optimisation (MDO) by Yang et al. (2001), collaborative optimisation by Braun (1996), multi-objective optimisation for engineering process by Suppapina et al. (2000) and Shea (2001) and multi-criteria optimisation in product platform design by Nelson et al. (1999). Product cost and lead-time are the most important parameters for a company to make a decision. They are indispensable for the success of a company. Hence, to optimise the product development process decisively, additional tools and methodologies for the optimisation of the PD process are required.

Instead of lead-time optimisation, the literature and research reports are more concerned with the optimisation of cost. Traditional cost estimation systems require input data, process sequences, processing times, hourly rate for labour or machines, and other miscellaneous data. Usually, these data are stored in a cost database. With these data in place, the estimating process simply breaks down to a series of simple calculation and data storage. For example, Winbourne and Toolisie (1991) proposed a cost estimating system that can be used at the design stage. By their costing system the detailed specification of the process sequence is not necessary for the cost estimation. A major drawback in their system is to require a user to estimate some initial certain parameters. No approach is given on how to estimate intelligently the values of these parameters. Wong et al. (1996) presented a cost estimating system, called the Totally Integrated Manufacturing Cost Estimating System (TIMCES), which integrates CAD and process planning techniques into a unified system. Input for this system consists of the required processes, the process sequences, and cost databases. However, the TIMCES still considers only predefined process sequences. Kamath et al. (1993) proposed a system to estimate the manufacturing cost of sheet metal parts in conjunction with their process plans. Based on the standard process sequence for the part family, the lowest production cost is found by selecting among the alternative machines in each process stage. Their system only considered 2-D shapes or sheet metal parts. Examples of the traditional cost estimating systems can be found in Casey (1987), Goldberg (1987) and Ostwald (1984).

These traditional cost estimation systems have the following drawbacks. First, these traditional approaches do not have a global optimisation algorithm that can optimise the design, process and manufacturing process. Second, the traditional systems are never integrated with a CAD system for
automatic data (cost and lead time) estimating. Last, the traditional systems are slow and time consuming. Under traditional cost estimating approaches, product costs may be misleading, performance measures inappropriate, cost control unfocused, and accounting systems irrelevant for cost management purposes. Yet accurate product costs are at the heart of pricing decisions for new product introductions, obsolete product withdrawal, and responding to competitive products.

From the foregoing discussion, it is clear that cost estimation is an area of great research interest. New technologies such as neural network (Ehrlenspiel and Schaal 1992, Bode 1998, 2000), activity based costing (Koltai et al. 2000), log-linear and non-linear learning curve model (Timothy 1999), AI and mathematical models (Xie et al. 2001b) have been widely used. Although several efforts have been made, there does not seem to exist a widely accepted system or a system with a wide applicability. In IRPD, not only the cost, but also other key parameters (lead-time, quality, etc.) are very important for products to compete in the global market and help to make earlier decisions. The product developed with lowest cost, for example may take longer and it may lead to losing the market. Thus, real time global multi-objective optimisation is vital for achieving RPD goals as the market changes very fast. To meet the requirements, the global optimising system should have following features: 1) optimisation of the key parameters (e.g. cost and lead time) through the whole life cycle of a product, 2) robustness, 3) automatic generation of process alternatives, 4) specification and representation of process alternatives, 5) the gaining of a wider acceptance by easy connection with commercial product development software packages, 6) adaption to the design, production, and manufacturing environment changes, 7) manufacturability verification, and 8) automatic selection of the process with optimal results. Further research that leads to a multi-objective global optimisation system with these features is urgently required by industry.

2.6.5 Virtual Reality for Manufacturing

To be able to consistently handle the demand for increased efficiency, quality, and flexibility, and to reduce the development lead time and cost, it is becoming increasingly important to structure the PD process by following a well-developed methodology. One important aspect of this methodology is the use of models and architectures that provide an abstract, simplified representation of reality (Doumeingts et al. 1995). This leads to virtual Reality (VR). VR can be used to make better decisions. Manufacturing processes can be defined, modelled, and verified (pre-run) before they can be actually implemented. For example, virtual NC programming and tool path control (Virtual NC), virtual robotics programming (IGRP), virtual instrumentation and shop floor control are the areas that have currently received considerable attention (Boman 1995). An improved decision-making process leads to increased efficiency and reduced costs, which are the two main goals for RPD. VR is an emerging technology with potential manufacturing applications in areas such as product design and modelling, process simulation, operating planning and real time shop floor controls (Stampe et al. 1993, Pratt et al. 1994).

In IRPD, VR plays an important role as it can be used to achieve the following functions: prediction of system performance, evaluation of a certain feature in the system, design verifications and tests, comparison between alternatives, gaining knowledge of the system at different life-cycle phases, problem detection, and presentation of predicted results. This research area covers topics such as distributed VR through the Internet, the integration of VR with other PD tools or methodologies, and the created VR manufacturing environments. Many researchers have made contributions in this research area. For example, Virtual Reality Modelling Language (VRML) has become an industrial standard (including the Internet version of VRML) and has been used in PD processes (http://3dgraphics.about.com/cs/vrmibrowsers/index.htm). There are a number of VR tools available in the commercial market (reviewed by Klingstam et al. 1999). Some VR tools such as QUEST, IGRP and virtual NC (Deneb Co.) have been developed and used in manufacturing. However, a VR manufacturing environment is a complex world which is made up of a large number of different 3-D models. There are problems in creating or adopting such an environment for the RPD process. These problems call for further research. First, it is a tedious and pains-taking process creating a virtual manufacturing world (Stampe et al. 1993 and Evans 1994). Second, the integration of the VR world with other PD tools is not simple. Last, the
speed of VR via the Internet has to be improved.

2.7 A Proposed IRPD System

The path to IRPD paradigm can be paved by re-engineering the enterprise, re-design and optimised PD process, and made more agile by using various enabling technologies. Through my research, an Internet based RPD system is suggested in this subsection for the rapid development of injection moulds. This system can be used under the heterogeneous design and manufacturing environments. To minimise the cost of building IRPD systems, it is preferable to standardize the IT infrastructure as much as possible through establishment of a software platform. This platform will be used to develop interoperability standards needed by the company to integrate the manufacturing processes. This proposed system does not intend to integrate all the Internet based systems as mentioned before in this chapter. It focuses on the key technologies that can be developed and integrated to support rapid injection mould development. This system has been tested in a mould manufacturing company in Christchurch. Although this system is proposed for rapid injection mould development, the structure of the system extends well for the development of other OKP products. In fact, an Internet based RPD system with a similar modular structure has also been developed (Xie et al. 2001b) for the rapid development of sheet metal parts. The effect of this research will enable the mould industry to gain the benefits from open, modular, and re-configurable integration of commercial software applications that support the design, planning, and production within the environment.

Figure 2.2 shows the reference structure of the IRPD system. It consists of: 1) an IRPD software platform, 2) a customer interface model that is developed based on QFD (quality function deployment) method, 3) an Internet based integrated product design environment (IPDE), 4) an Internet based product design environment for supporting product design, 5) an Internet based virtual process planning/assembly environment, 6) an Internet based virtual simulation environment, 7) an Internet based virtual manufacturing environment, 8) Internet based design/manufacturing product knowledge bases, and 9) a Global Cost Optimisation Model.

It is obvious that the IRPD system can be accessed from anywhere either inside or outside of a company. This will first help the company’s employees to work in an integrated information environment. An Internet based integrated product data environment is built for data communication and exchange.
between various systems. The structure of the Integrated Product Development Environment (IPDE) was well addressed by Tu et al. (2001) and the information framework and data models have been well explained by Tu et al. (2000a and 2001). They can share the PD software tools, data, and work concurrently to develop a product. And this system brings together expert knowledge of various fields in the early phase of PD. The meetings and discussions through a PD cycle will be automatically recorded by the system when they communicate through the software platform. Second, the company’s marketing people can access this system via the Internet to get the on-line engineering support from the company to address, discuss and meet customers’ needs. In the system, the modules and technologies are integrated into a continual process chain. This process chain comprises all phases of PD process, from the customer requirements, first CAD draft and finally to the market. The system is equipped with all the necessary technology for a fast and cost-efficient development of injection moulds. Tools like Internet communication, product data management and computer simulation are integrated into the continuous flow of data. The company’s existing computer aided engineering and management systems such as CAD systems, knowledge management systems, and database management system can be integrated through the software platform of the proposed IRPD system.

The function of the key modules in the system has been well addressed by Tu (1996, 1998, and 2000a) (as shown in Figure 2.3). In this system, a product can be designed, planned and virtually manufactured and tested with the applications of virtual technologies and methods by using the computer simulations. The customer interface model is defined to manage dynamic customer requirements so that the IRPD environment reacts more rapidly to the changing requirements of the market. Through the model, customer’s voices are gathered through provided Internet based customer interfaces and transferred to a shared QFD database (Tu and Xie 2001). A data analysis model is defined for matching the customer’s requirements to engineering requirements. The data/knowledge are recorded in the Design/Manufacturing database after the product is developed. They can be further used as knowledge or references for similar product development. Reuse of the past experiences and knowledge can greatly shorten the PD cycle time. An Internet based DFM system is developed for improving product and manufacturing performance. An Internet based DFA system is developed to cut down assembly time and cost. A cost optimisation algorithm based on the shortest insert algorithm was proposed for reducing PD costs (Xie et al. 2001b). As can be seen from Figure 2.3, the IRPD system integrates software tools for correspondent PD process. Hence, with the integration of these technologies and models, the goals of the rapid mould development, i.e. shorter lead-time and cheaper cost, can be achieved through the system integration and knowledge reuse.

Figure 2.3 the structure of IRPD environment
2.8 Chapter Conclusions

This chapter overviews the recent approaches of Internet based PD methodologies for the purpose of rapidly and economically developing customised or OKP products. Although quite a lot of literature or research reports were published in journals and some Internet based PD systems were reported to be implemented in industry, to my knowledge a paper which gives a systematic overview and summary of these research and industrial implementations has not been found. This chapter aims to fill this gap. As the manufacturing companies, particularly those OKP companies are rushing to acquire the Internet based PD systems for the purpose of production integration, effective communication, resource sharing, increasing productivity and flexibility, and reducing costs and rework, some research and development directions have been identified in this chapter based on the existing research review and industrial needs. At the moment, the research and implementation of Internet based systems are far behind industrial expectation. On the other hand it is necessary for industries to understand the development conditions and currently available technology limits for developing the IRPD system so that they can properly select and adopt the suitable strategy of developing the IRPD system.
CHAPTER 3

AN INTERNET BASED INTEGRATED PRODUCT
DEVELOPMENT SYSTEM FOR RAPID SHEET METAL
PART/PRODUCT DEVELOPMENT

To further develop the reference system structure of the Internet based integrated product development system, outlined in chapter 2, into a working product development system, this chapter will propose an integrated product development system for rapid development of sheet metal parts. The structure of the system will be discussed and case studies will be carried out to test the idea of the Internet based integrated system. In this chapter, several major modules of the system are discussed. Mainly these modules include the structure of the integrated product development system, an integrated data environment, an RTCAPP (real time computer aided process planning) module, QFD based global customer interfaces and design/manufacturing knowledge bases for supporting product design and manufacturing. This chapter explores the definition and the structure of the system for rapid development of sheet metal parts.

3.1 Introduction

Nowadays companies in the sheet metal industry are under great pressure due to the globalization of the market, the short life cycle time of products, increasing product diversity, high demand for quality and short delivery times. To be competitive in the market, sheet metal companies must gain enough production flexibility to produce rapidly various sheet metal products with acceptable quality. Multinational sheet metal corporations especially, must use new technologies for their product development, re-engineer their organisational structures and enhance their ability to make decisions correctly through the whole product development cycle, particularly at the early product design stages. These new technologies include internet technology, CAD/CAPP/CAM integration technology, computer simulation technology, rule based reasoning (Dixon 1986), constraint-network (Young et al 1992), knowledge bases and optimisation theory (Dowlatsahi 1992).

Usually, sheet metal product design and manufacturing processes planning are carried out in a series of stages using different software packages, especially for bent sheet metal products. First the product is designed by using a 3D CAD system, usually using the wire-frame representations. Then an unfolding software package is used to extend the model into the 2D drawings of the flat parts. A computer-aided process planning (CAPP) system is employed to plan manufacturing processes and select the necessary tools. To minimize the production waste and to improve the production efficiency, optimisation software systems (e.g. optimal nesting software systems, path-planning systems) are used. CAM software systems are also often used in a sheet metal manufacturing company to generate NC programs for CNC sheet metal processing machines, e.g. a punching, shearing, or folding machine. Before delivering the NC programs to a CNC sheet metal processing machine, a simulation software package is used to simulate the processes and consequently to modify or refine the NC programs as well as to adjust the tool paths. Usually, in a sheet metal product development cycle, these processes are carried out separately and sequentially in different branches of the company. With this traditional design and manufacture method, a sheet metal product is frequently designed without a systematic consideration of down-stream product development requirements, such as process planning, manufacturability, production scheduling and manufacturing optimisation. Also, the feedbacks from these down stream processes to the product designer are only made after the product is designed or even
manufactured. This often results in expensive and time consuming rework. Consequently, it affects the quality, cost and delivery time of the product.

To fill the gap between the design and other downstream product development processes, quite a few research projects have started to focus on the concurrent product design and manufacture and intelligent decision support in earlier product design stages (Kusiak 1993, Prasad 1996, Bartolotta et al. 1998, Dong et al. 1998, Tu et al. 2000a). Concurrent product design and manufacture means that a designer should consider all possible requirements from the down stream product development processes and incorporate them into the product design. However, there is no platform or system available that can be directly used to support concurrent sheet metal design and manufacturing process planning by different branches or partners that may be globally distributed.

An integrated product development system that can support the intelligent decision process in earlier product design stages, can be realised by applying advanced computer technologies, such as computer aided design and engineering technology (e.g. ProEngineer), manufacturing optimisation software packages (Dowlatsahi 1992), computer simulation technology, and knowledge base technology (Bartolotta et al. 1998). By using these computer software packages or systems, a designer can simulate and virtually observe the down stream processes for a possible product design on a desktop computer. This integrated product development system can help the designer to understand the down stream process requirements and to design a product right at the first instance.

However, simply putting together several computer aided sheet metal design and manufacture software systems does not provide a sound overall solution for achieving concurrent engineering in a modern sheet metal manufacturing company as its “partners” are globally distributed. A main problem to realising concurrent engineering in such a company is providing intensive real time information interchanges (Devarajan et al. 1997) and optimisation (Xie et al. 2001a) in the company. Also, the complexity of a sheet metal product produced by the company makes it impossible for a single designer to manage the complete product development process. Successful development of such a complicated sheet metal products needs the contributions and expertise from almost all the people in the company. On the other hand, the various computer software packages or systems are often developed by different vendors and some software packages are not suitable for their special requirements. Hence to achieve concurrent design and manufacturing of sheet metal parts, a system that contains several application modules to meet special requirements and can support efficient and effective information interchange among different software systems, different departments and different people in the company, is very important. Correct and effective information exchange throughout the product development life cycle can shorten the product development lead time, improve the production efficiency, achieve high quality, and, at the same time cut down the production cost.

The purpose of this research is to develop an integrated product development system for intelligent concurrent design and manufacturing of sheet metal parts. The structure of the integrated product development system will be addressed in section 3. To support the development of this system, several major modules will be discussed in the following sections, e.g. an Internet based integrated data environment to support information (data) sharing, QFD based global customer interfaces and a WWW based knowledge bases module to support intelligent concurrent sheet metal product development.

3.2 The Integrated Product Development System for Sheet Metal Design and Manufacturing

Because of the industry’s increasing needs for other types of information or knowledge in sheet metal design and manufacturing process, new classes of tools to support knowledge-based design, product data management and concurrent engineering are urgently needed in some companies. However, traditional CAD, CAM systems usually run independently and the information transfer among these systems is through certain file format (STEP, DXF, etc.). Delays, errors and long procedures of product definitions are also caused by lack of supporting information. Also, traditional CAD, CAM tools mainly focus on database-related issues
(Simon et al. 2000) and do not place a particular emphasis on real time information modelling for product representation in an integrated system.

Owing to the numerous and complicated interactions among the tasks of concurrent sheet metal product development processes, an Internet based integrated sheet metal product development system is proposed as shown in Figure 3.1. The integrated product development system reported in this chapter aims to solve these problems. This system is based on a WWW platform that can be used by most operation systems. This system contains several modules, i.e. an unfolding module, an Internet based data integration platform, design/manufacturing knowledge bases, data communication tools among different modules, a RTCAPP module, a CAD module, a CAM module, a cost estimation module, computer simulation platforms, graphical user interfaces (GUIs) etc.

![Figure 3.1](image)

**Figure 3.1** The integrated product development system for sheet metal concurrent design and manufacturing

The unfolding module in this system adopts zero thickness and zero bend radii principles in order to facilitate the bent sheet metal part design process. The data integration platform is a data integration environment for product design, unfolding, process planning and manufacturing process, which is developed based on the information integration framework, Pro/INTRALINK and the API of Pro/Engineer (Xie and Tu, 2000). The data communication tools are developed for information exchanging among different systems, i.e. the data communication tool between the integration platform and CAPP software provides CAPP with suitable parametric data of the sheet metal parts and sends all this information back to the data integration platform in real time. This information, e.g. as process plans, manufacturing parameters, tool sequences and cost, can be used by other modules. The data communication tool between the integration platform and CAD software captures related information from knowledge bases and provides CAD with suitable information such as cost, unfold ability, etc, so that a decision can be made.

The customer interface module is provided to satisfy the demands of the customer. This is crucial these days to market competitiveness and business success, and this trend is set to become one of the prerequisites for business survival in this century. Organizations that manage to meet or improve on customer requirements will emerge as the leaders in the global market. Over the last decade, QFD has developed into a proven
quality management concept and methodology which can facilitate the understanding and response to customer requirements at various stages of product design, development and manufacturing with effective allocation of enterprise resources.

The knowledge bases aim to provide the means to support human decision making on both earlier design process and later design and manufacturing process by capturing and making available different types of design and manufacturing knowledge. The knowledge bases are built based on a "four-layer" structure of the information framework and are saved as a part of the company’s overall database. There are several knowledge bases involved in this project, e.g. tool knowledge base, design knowledge base and manufacturing knowledge base.

The Internet based cost estimation module aims at the product development cost estimate and optimal control under the complexities of a global manufacturing environment. The details of the cost estimation and optimisation algorithm will be addressed in Chapter 6. This system is able to provide in-time cost estimates and optimisation results to product developers as important decision-making references in a concurrent engineering approach of the product development. This concurrent approach enables product developers to design the product, plan the processes, schedule operations on the shop floors, manage logistics, and estimate and optimally control the cost feature by feature. This system can help the product developers to clarify the product development life cycle data at earlier product development stages and hence avoid rework that often occurred in the traditional sequential approach to product development. Avoiding rework in a product development cycle, particularly for customized product, will reduce the product development cycle time and cost significantly.

The RTCAPP module contains several sub modules, including knowledge based auxiliary path automatic insertion, an optimisation module, a database, a tool library and a module library. This module will be explained in Chapter 5. The optimisation module is the most important module for achieving efficient and economic manufacturing. It includes the tool sequencing, tool path optimisation and nesting optimisation of sheet metal parts with complicated shapes, etc.

Graphical user interfaces are also needed to serve as a bridge between the integrated product development system and human users. Creating GUIs for this integrated product system is very important and it is largely a matter of describing where every interface component should appear on the screen, how it should appear and what should happen when the user interacts with it. The GUIs are designed by using several platforms, and they can be viewed through Web Browser or Internet Explorer. These interfaces are developed not only for information or knowledge sharing among different systems and users, but also for the “appointed” user to define suitable data structures for knowledge bases and information objects according to different applications.

It is obvious that this integrated product development system can be accessed from anywhere either inside or outside a company. This will first help the company's employees to work in an integrated information environment. They can share the product development software tools and data, and work concurrently to develop a product. The meetings and discussions through a product development cycle will be automatically recorded by the system when they communicate through this system. Second, the company’s marketing people can access this system via the Internet to get the on-line engineering support from the company to address, discuss and meet customers’ needs.

In order to achieve this system, this chapter will discuss some key modules including the Internet based Integrated data environment, QFD based customer user interfaces, simulation platforms and knowledge bases to support intelligent concurrent sheet metal product development. The zero thickness and zero bend radius principle and the information integration framework will be discussed in the next chapter. These modules will be discussed in following sections.
3.3 The Internet based Integrated Data Environment

Data sharing in the RPD process in such a heterogeneous environment has always been a difficult task. For example, different design, analysis and planning tools, such as geometric modellers, analysis tools, planning tools, CAM tools, are all critical components of the overall process of product development. A fundamental problem with data sharing across these tools is that the output of one tool is often required as the input of another tool. Dynamic information modelling and exchange is involved in the product development process. Also, as different tools have different data formats, it is therefore difficult to move data from one step of the design process directly to another step without data translation. Hence, an efficient Internet data environment that can perform these functions will be a key for supporting the development of the product development system.

An Internet based integrated data environment is proposed for the sharing of product information across diverse software tools in the Internet environment. This Internet based data environment aims to build an information bridge to fill the gap between the sheet metal part design, process planning, simulation and manufacturing systems. It is an important issue in supporting integrated and concurrent product development. With the development of following technology and methodologies, the data environment that can be used to build information models for information sharing among different software packages can be built.

The STEP plays an important role in supporting the functionality of the Internet based integrated data environment. In order to maximize interoperability with most of the existing software packages used in industry, the framework makes use of standard representations when possible. For the representation of geometry, this project uses ISO 10303 that is more commonly known as STEP (the Standard for the Exchange of Product model data, officially ISO 10303) (ISO 10303-11:1994E). STEP is a family of standards specifying a description of product data throughout the product’s life cycle in a computing-platform-independent manner; each individual standard in STEP can be used to define the information requirements for a particular area of design, manufacturing, engineering, or product support (ISO 10303-11:1994E). The conceptual schemas of different Application Protocols (ApS) within the STEP standard provide the semantic basis for the data managed by the integrated data environment, where an AP provides a standard schema definition for the data that is used within a particular design and/or analysis domain. We have also adopted and modified the STEP concept of Units of Functionality (UoFs) to define our notion of conceptual sub-components of exchange files (Liang 1997).

As the representation of non-geometric information is at a far more preliminary stage (Wang et al. 1995) and industry’s reliance on non-geometric information and knowledge-based design and manufacturing increases, new modelling methodologies for design and manufacturing knowledge such as function, behaviour, knowledge and other kinds of non-geometric data are much needed. The use of UoFs and relationships will help to create data sharing information models for real time data communication among the different software tools. This issue will be discussed in greater detail in Chapter 4.

For the consideration of the STEP and the working environment of the product development system, an integrated global data structure is the key for building the Internet based integrated data environment by using STEP standards. This issue will be discussed in detail in Chapter 7. The emphasis of the global data structure is placed on the integrated data management and reuse of past PD experience to support a company’s aim to shorten its product development cycle. The integrated data structure was modelled by the EXPRESS of STEP. In terms of this data structure, a design/manufacturing knowledge base was developed as a major part of the WWW (world wide web) based information management system. The basic principles and concepts of the knowledge base and the WWW based information management system will be presented in Chapter 7.

3.4 QFD based Global Customer User Interfaces

Satisfying the demands of the customer is crucial for market competitiveness and business success these days, and this trend is set to become one of the prerequisites for business survival in this century.
Organizations that manage to meet or improve on customer requirements will emerge as the leaders in the global marketplace. Over the last decade, QFD has developed into a proven quality management concept and methodology which can facilitate the understanding and response to customer requirements at various stages of product design, development and manufacturing with effective allocation of enterprise resources.

The voice of customer (VoC) or customer requirements / attributes are often expressed in linguistic terms which are usually non-technical in nature. Sometimes, it is difficult for engineers to translate these requirements into definitive product and engineering specifications. Quality Function Deployment (QFD) (Akao 1990, Cohen 1995) is a well-known quality management methodology that helps channel customer requirements through various stages of a manufacturing cycle, such as design, part planning, process planning and operation planning on a product. The process can be represented graphically by a series of inter-connected structured matrices called the Houses of Quality (HoQ) (Hauser and Clausing 1988). A typical HoQ will normally consists of eight parts, i.e. voice of customers (or customer attributes), voice of engineers (or product attributes), correlation between product attributes, competitive benchmarking, relationships between product attributes and customer attributes, priority of product attributes, target design values, and concept evaluation.

The semantics in the voice of customers may contain ambiguity and multiplicity of meaning (Khoo and Ho 1996), and it makes the interpretation of VoC into precise product / engineering attributes rather complicated and difficult. Besides, most of the recent researches in QFD assumed that the customer and engineering / product attributes are crisp, complete and independent of one another, though it is seldom the case in reality. These over-simplifications can lead to discrepancies in design priority setting and unbalanced resource allocation. Hence, a conventional HoQ can only give a picture of how the customer attributes could possibly be met, and indeed many companies apply QFD and HoQ to do just that. However, in the proposed Customer Interface Model, QFD is just the beginning of a series of inter-linking processes and planning stages. In this model, a focused HoQ is used.

The structural framework of the focused HoQ is similar to that of the basic HoQ, however more detailed quantitative analyses would be conducted to establish the information it holds. First of all, a dictionary of key words abstracted from a company’s VoC records has been developed to standardise the VoC and to clearly identify the customer needs. Affinity Diagram was applied to regroup the more essential customer attributes. Pairwise comparison exercise was carried out on the customer attributes, and the findings could be further analysed using the Analytic Hierarchy Process (AHP) (Saaty 1990, 1994). The resulting Relative Weights of Importance for the customer attributes are normalised and expressed in percentages. A dictionary for voice of engineers was also developed to standardise the technical languages used in the company.

One of the distinct merits in this focused HoQ lies in the fact that the attribute relationships are established quantitatively, instead of relying on qualitative interpretation. The attribute relationships are worked out based on the contributions and thus the importance of individual product attributes towards the fulfilment of each customer attribute. AHP plays an important role in the quantitative analysis in the focused HoQ and in the calculation of the Relative Weight of Importance for each product attribute.

The Target Design Values in QFD represent the design specifications for individual product attributes used to guide subsequent design and production activities. The output from this Customer Interface Model is a list of product attributes with relevant target design values, i.e. features and the design specifications of a product required by the customer. These features will be classified into primary features and secondary features according to the design and manufacturing requirements or limits.

To design QFD based customer interfaces for rapid development of sheet metal parts, case studies have been carried out in a sheet metal manufacturing company (Kam 2000). A Complete Customer Database and an Engineer Database have been created as the input of QFD process. A relationship matrix between VoC and Voice of Engineer (VoE) has been created using Microsoft Excel. In order to allow the link to takes place,
an interface is designed. An interface is a program that enables the user to input data into it. The source code of the QFD based customer interface is attached in Appendix 2.

3.5 Simulation Platforms

Some advanced computer simulation packages can be used as effective tools in the integrated product development system to approve some critical manufacturing processes and to derive the operational cost for manufacturing a product. A simulation platform for sheet metal cutting and punching is developed by the author (Xie 2001a). The CNC machines that will be used to manufacture certain sheet metal parts can be built by using VNC simulation package from Deneb Co. Ltd. Those machines are saved in a library and can be selected by users through the simulation platform. The VNC is one of the four simulation products of Deneb Inc., USA. It is developed for simulating all kinds of CNC machine tools, and it is a 3-D simulation package. In the simulation platforms, a part can be machined just like being machined on a real CNC machine. By using this simulation model to virtually machine a sheet metal part, some design mistakes and hidden production reworks can be avoided, e.g. for machining a mould with complicated sculptured surfaces (Lee et al. 1993) some dead cutting points due to the cutting tool collision can be detected. Through the simulation platform, the detailed times for manufacturing a product in the physical shop floor can be observed and recorded. These include machine times, set-up/tear-down times, transportation times, waiting times, machine broken times, and idle times. Based on these times, the cost for manufacturing a product in the shop floor can be derived. Furthermore, a production schedule for producing a new product can be traced out through manufacturing this new product together with other products being produced in the company. Details of the simulation platform and other key application modules in the platform will be explained in the following chapters with case studies.

3.6 Knowledge bases to Support Intelligent Design and Manufacturing

The design/manufacturing knowledge base for sheet metal parts can be built as an on-line data library or knowledge base to support all kinds of activities through a product development cycle. The data structure of the design/manufacturing knowledge base can be modelled according to the sheet metal part and it is related to the information flow in its development process such as resources, tool information, etc. After one product has been developed, all the related information can be stored in the design/manufacturing knowledge bases and can be retrieved by data searching tools for use with similar product development. The products in the design/manufacturing knowledge bases represent the company’s existing products. They are modelled and recorded in the data format according to the data standard provided by the integrated data environment. Other information objects or activities have been modelled from a database approach (Figure 3.2), i.e. the suppliers’ information model includes all information about the company’s suppliers, i.e. contact information, product specification, reference prices, delivering lead times, reputation assessment and quality assessment. The inventory information object can also be modelled and used to record the company’s present inventory level, which is used as a part of the company’s MRP system. The tool information objects including its relationship with certain features of sheet metal product are modelled and saved to the database as the company’s tooling information. The resource objects are built and used to keep the records of the company’s equipment, such as cutting machine, punching machine, compound machine, etc., employees and others. The present and previous data are also recorded in design/manufacturing knowledge bases. Based on the objects and relationships among objects, knowledge or useful information can thus be formed to support the development of new products.
Chapter 3 An Internet based Integrated Product Development System for Rapid Sheet Metal Part/Product Development

Based on the object-oriented modelling concept, the relationships between features and knowledge units are established by defining methods for knowledge activation. When the method or function is activated, a corresponding knowledge object is instanced to perform certain operations on the feature through running rules belonging to the knowledge object. Similarly, data are tightly associated with knowledge objects through defining methods or function for activating relevant data objects from knowledge objects.

To illustrate the dynamic aspect of interactions among feature, knowledge and data, the punching tool sequencing model of sheet metal compound manufacturing (cutting and punching) is used as an example

Information-Models for design/manufacturing knowledge bases;

ENTITY Product;
  Product_family : STRING;
  Product_Title : STRING;
  ProductDescrip : STRING;
  Drawing_ID : STRING;
  DesignfileNo : INTEGER;
  Productassembled List : LIST[0:?] OF DesignVerification;
  Process_plan;
  SalesInformation : SalesOrderForm;
  Toolmanufacture : Tool;
END ENTITY;

ENTITY SalesOrderForm;
  Salesordercode : STRING;
  OrdereDate : Date;
  DeliveryDate : Date;
  Customer : Customer;
  ProductCost : STRING;
  LeadTime : STRING;
  PlacedBy : Employee;
  ItemList :LIST[0:?]OFOd;
  OrderQuantity : INTEGER;
  OrderDescrip : STRING;
  Status : TRUE/FALSE;
END ENTITY;

ENTITY Tool;
  Tool_code : STRING;
  ToolTrailNumber : STRING;
  StaffCode : STRING;
  ToolRoomManager : Facility;
  SetupTime : INTEGER;
  OperationTime : INTEGER;
  ToolShape : INTEGER;
  ToolMaterial : STRING;
END ENTITY;

ENTITY Supplier;
  SupplierCode : INTEGER;
  SupplierName : STRING;
  Product : STRING;
  Contact : STRING;
  TelNumber : INTEGER;
  FaxNumber : INTEGER;
  ContactPerson : STRING;
END_ENTITY;

ENTITY Resource;
  ResourceId : STRING;
  ResourceName : STRING;
  ResourceDescription : STRING;
  Supplier : Supplier;
  ResourceCost : INTEGER;
  Location : STRING;
END ENTITY;

Figure 3.2 a selection of entities from the EXPRESS schema for the knowledge bases design

(Champati et al. 1996). For example, once the user enters this module, the tool process planning knowledge object is instanced to have the user select a tool from the tool library. Feature dimensions are instanced by feature objects; a tool object will be available after the user chooses a tool. The tool sequencing module will automatically analyze if it can be used to punch the feature, and the tool-sequencing object will return certain information to show the status. The relationship between tool object and feature can be created, and can also be used as knowledge for the next tool selection of the same feature. The design and manufacturing knowledge bases can also be built in the same way.

The author has developed a platform for storing knowledge by using the object-oriented modelling language, as well as tools for examining, manipulating, searching and retrieving knowledge bases, i.e. the
information access tool and collaborative communication tool have been developed as tools for efficient and effective knowledge access so that the knowledge bases can be used on a large scale in a distributed environment (i.e. over the internet). This WWW based database/knowledge base system will be addressed in Chapter 8.

3.7 Case Studies

This section mainly discusses case studies on application software packages developed by the author, e.g. a CAD/CAPP/CAM system (Xie and Tu, 2000), a simulation software package, a shortest non-machining path optimisation software package. Also, some application modules of the integrated product development system are briefly discussed.

3.7.1 Programming Tools

The programming languages that have been used to achieve this integrated product development system include Visual C++, VBSCRIPT, JAVA SCRIPT and EXPRESS. Visual C++ is an object-oriented (O-O) language, which supports class and object operations. It is used for developing most of the application modules and tools. VBSCRIPT and JAVA SCRIPT are used for design/manufacturing knowledge sharing and client-server communication. JAVA SCRIPT is a cross-platform object-based scripting language for client and server application and can be used to create applications that run over the Internet. Also, these languages make it easy to achieve all the application modules of the integrated system.

3.7.2 Application Modules Test

In order to achieve the integrated product development system for sheet metal intelligent concurrent design and manufacturing, several modules have been developed. This chapter will discuss the application results of some of the major modules with case studies, i.e. manufacturing cost estimation module, WWW based design/manufacturing knowledge bases and a RTCAPP module including several sub modules, e.g. path planning, nesting, knowledge based auxiliary path inserting. Some of the modules will be further addressed in the following chapters.

The manufacturing cost estimation module estimates the manufacturing cost of the sheet metal part. This module takes the input from the NC programming and the knowledge based CAPP modules. The cost object includes several elements. These elements are used to calculate the total cost of the product, and the machining time is calculated from the virtual simulation platform. The set-up times are estimated times associated with a particular operation. The cost of using different manufacturing methods can be calculated from machining times and expenses, i.e. for a flame cutting processes, the manufacturing cost can be calculated if the manufacturing time and expenses (cutting gas, auxiliary gas) are known. The cost of sheet metal utilization ratio can be calculated according to the data input from the nesting optimisation algorithms sub module. The total manufacturing cost of the part can be calculated by adding the estimated material cost to the sum of the costs of all processes. As one can see, all the optimisation algorithms have great influences on cost. The function of cost estimation has been defined in the data integration platform, which can be sent to CAD or other modules.

The RTCAPP module in this chapter presents a knowledge based RTCAPP system based on the integrated system, which includes several optimisation modules that aim to improve the manufacturing efficiency and cut down costs and knowledge based auxiliary paths and manufacturing parameter selection for sheet metal cutting processes. Chapter 5 will give the detailed definition and how to achieve the RTCAPP module. The functions in the RTCAPP system are designed to achieve different application tasks. These functions can access all required information through the data integration platform. Tool database, cutting parameter database and process rule database are all connected with the data integrated platform through ODBC. Functions have been defined for automatic tool selection, tool path optimisation, nesting optimisation, manufacturing parameters, etc. For the sheet metal manufacturing RTCAPP system, real time
optimisation modules are very important for the manufacturing efficiency and cost, while manufacturing parameters and auxiliary path will influence the manufacturing quality of sheet metal parts.

All the key modules and tools have been tested in this system, i.e. a communication tool with CAD (Pro/Engineer), CAM, knowledge capturing tool, shortest manufacturing path optimisation, nesting optimisation, etc. Figure 3.3 shows some case studies of the integrated product development system. Figure 3.3 (1) shows the result of a simulation platform for shortest path optimisation, process planning and nesting algorithm. This system takes the geometric parameters as input and simulates the manufacturing process while choosing different CNC punching machines, bending machines and cutting machines. In order to improve the utilization of sheet metal parts and manufacturing efficiency, a RTCAPP module has been developed. This module contains a nesting optimisation, auxiliary path inserting and a shortest non-machining path-planning module. Figure 3.3 (2) shows the output from the RTCAPP software package that has been developed by the author. The test results in several sheet metal companies show that the system can greatly shorten product development time, cut down cost and improve machining efficiency. Figure 3.3 (3) shows a WWW based QFD based customer interface, which can be used by product designers to make decisions in the earlier design stage. This product knowledge view platform is supported by knowledge bases that are built based on the information data environment. Knowledge mining tools have been developed in order to search and locate related knowledge. With the support of product knowledge, it becomes easier for product designers to make decisions at earlier design stages. Figure 3.3 (4) shows the simulation results of the shortest path planning optimisation module. Different optimisation algorithms have been used and tested in
this system. Figure 3.3 (5) shows the cost estimates for a certain flame cutting method by using three different optimisation algorithms, when given information, such as cost of using a cutting method, tool set-up and exchange time and cost, etc. The utilisation ratio of a sheet metal part can be calculated by the nesting algorithm in real time, so costs can be optimised and can be sent back to the designer through the communication tool, hence economical design can be achieved.

3.8 Summary of the Chapter

In order to rapidly develop sheet metal parts, an Internet based integrated product development system has been put forward in this chapter. An integrated data environment that aims to provide an information methodology to support sheet metal parts intelligent concurrent design and manufacturing has been put forward and explained. This integrated data environment can also be used in other areas for building knowledge bases to support product design and manufacturing. Intelligent decision support is achieved at an earlier design stage through this system by correctly defining the structure of the company's knowledge bases, which includes the representation, capturing, sharing and reuse of corporate design and manufacturing knowledge.

The advantages of the integrated product development system are:

1. Establishment of a platform-crossed client/server management pattern. The distributed team members can share in product design and manufacturing knowledge in real-time through the uniform graphic user interface;

2. The development of a knowledge-based intelligent decision support platform in the earlier stage of product design;

3. The development of the integrated product development system that can support sheet metal parts concurrent design and manufacturing in a distributed computing environment through information integration;

4. It is multi-vendor software compatible.
CHAPTER 4
AN INFORMATION FRAMEWORK FOR RAPID SHEET METAL PART/PRODUCT DEVELOPMENT

Information integration is an important issue in supporting integrated and concurrent product development. This chapter explores the definition and the structure of an information framework for rapid development of sheet metal parts. This framework aims to build an information bridge to fill the gap between the sheet metal part design, process planning and manufacturing systems. It is based on the principles called zero thickness and zero bend radius principles, which are used to abstract the geometry entities of sheet metal parts in order to facilitate part modelling and information modelling. In this chapter, a tree based step-structure information modelling methodology for sheet metal parts has been put forward and a case study has been given.

4.1 Backgrounds and Motivation

Today’s manufacturing industry greatly relies on computer technology to support activities through a product’s life cycle. Effective and efficient information sharing and exchange among computer systems have been crucial issues. An information modelling methodology that describes information flow unambiguously throughout the product development life cycle is an enabling technology that facilitates the development of a large scale, networked, computer environment that can support RPD.

For the development of sheet metal parts, traditional product design and manufacturing processes are carried out in a series of stages, especially for bent sheet metal products. With this traditional design and manufacture method, a sheet metal product is frequently designed without systematic consideration of downstream product development requirements, such as process planning, manufacturability, production scheduling and manufacturing optimisation. Also, the feedbacks from these downstream processes to the product designer can only be made after the product is designed or even manufactured. This often results in a lot of expensive and time consuming reworks. Consequently, it affects the quality, cost and delivery time of the product.

To fill the gap between the design and other downstream product development processes, efficient and effective information interchange among different software systems, different departments and different people in the company is critical for concurrent design and manufacturing of a sheet metal product. Correct and effective information exchange throughout the product development life cycle can shorten the product development lead time, improve the production efficiency, achieve high quality, and, at the same time cut down the production cost.

In order to achieve intensive information interchange on sheet metal design and manufacturing processes and intelligent decision support in the earlier design stage, an integrated product development system where different software packages and people can share information simultaneously has been presented in Chapter 3. To support the development of this system, an information framework that allows the use of information in a shared environment is proposed in this chapter.

The information framework is used to build information models for information sharing among different software packages so that these software packages work in an integrated environment concurrently.
Also, the framework can be used for building design and manufacturing knowledge bases, which can be used as a reference for intelligent decision support. This work has been used to build a WWW based intelligent design and manufacturing knowledge bases and has been tested in a sheet metal company.

4.2 The Role of the Information Framework

Owing to the numerous and complicated interactions among the tasks of concurrent sheet metal product development processes, an Internet based integrated sheet metal product development system was proposed in Chapter 3. The platform contains several modules, i.e. an unfolding module, a Internet-based data integration environment, knowledge bases, data communication tools among different modules, a CAPP module, a CAD module, a CAM module, a cost estimation module, a computer simulation platform, GUIs, etc.

In this system, the relationships between the information framework module and other modules are shown in Figure 4.1. From Figure 4.1, the information framework provides feasible solutions to build automatic links between data of product design, process planning, unfolding software package, logistics management (i.e. supply, partner and sub-contractor chains management), CAM and costing. Product data models can be built to manage information flow among these application software packages. The fundamental problem with data sharing among these software tools is that the output of one tool is often required as the input of another tool, yet each system uses proprietary data formats that are not easily transferable to other systems. It is difficult, therefore, to move data from one step of the design process to another step without performing data translation. Past solutions have been focused on building translators between individual tools. For N different software tools as shown in Figure 4.2, this approach needs \( o(N^2) \) data translators. This is obviously not a practical solution to the problem of data exchange. For example, extensive real time data exchange between different systems via the Internet will cost more time and thus greatly influence the speed of Internet communication. Furthermore, it is important to the history of the design process to provide more than just a translator to move data from one tool to another. It is also
important to record different versions of design files, as well as the relationships that exist between the data in such files. In some cases, some tools may require combined input from several different tools. Hence, a solution as shown in Figure 4.2 (b) would be ideal to resolve the issues, where N different design and analysis tools only require O(N) translators, and information communication occurs through some common, shared data source. Such a solution depends heavily on standards for data exchange combined with the use of database technology for managing exchange files and data relationships.

This chapter presents the results with the development of an information framework. The information framework can be used to solve the problem of information modelling and real time exchange between different software tools in the Internet based concurrent and integrated product development system. This chapter will discuss how to build an information framework to support information/data modelling in the Internet environment.

4.3 The Zero thickness and Zero Bend Radius Principle

This section will discuss a generic handling methodology of bent sheet metal parts, as it plays an important role in product design and modelling processes and has been used in this research.

This research considers a generic handling methodology for its parametric shape of bent sheet metal parts. The details of the generic handling principles are:

- Zero thickness principle: supposing that the thickness of bent sheet metal parts is zero, designers do not need to consider the thickness of the parts.

- Zero bend radius principle: supposing that the bend radius is zero, that is to say, one does not need to consider the changes of parametric shape in the bend process.
Chapter 4 An Information Framework for Rapid Sheet Metal Part/Product Development

The above principles are based on the principle that the length of the middle layer of the sheet metal parts is the same before and after the bend process. The middle layer will change when the following situations change, e.g. product materials, bend method, and mould structure. A database was built for selecting the position parameter of the middle property layer (Zhou 1989). As shown in Figure 4.3, the length of the sheet metal part after unfolding can be calculated as follows (Liu 1998, Wang 1991, Zhou 1989):

\[ L = L_1 + L_2 + \frac{\alpha \cdot \pi}{180^\circ} (r + k \cdot t) \]

Where: \( L_1, L_2 \) are the length of two linear side;

\( \alpha \) is the bend angle (degree);

\( k \) is the position coefficient of the middle property layer;

\( r \) is the bend radius of the inner surface (mm);

\( t \) is the thickness of the sheet metal part (mm).

With these two principles, the bend parametric shape of sheet metals can be considered as a group of continuous zero-thickness planes connected with each other. For example, Figure 4.4 (1) can be expressed as Figure 4.4 (2) without consideration of the thickness of the sheet metal parts. The software will consider the part thickness and bend radius automatically.

The advantages of using the zero thickness and zero bend radius principle to handle the parametric

![Figure 4.3 the position of the middle property layer](image)

![Figure 4.4 a generic handling method of bend sheet metal part](image)

(1) A sheet metal part example

(2) The unfolded example sheet metal part
shape of bent sheet metal parts are as follows:

- Simplicity: a complicated 3D product model can be simplified into a 2D model. Hence complicated hand calculation can be avoided, which makes the information modelling process easier.

- At the design stage, one does not need to consider the plastic deformation of the products on the bend process (the software will automatically consider it).

- One does not need to consider parts thickness at the modelling process and 3D entities can be transformed to 2D plane entities with relationships.

The method of using the zero-thickness and zero bend radius principles to handle the parametric shape is for the convenience of the product designers. This generic handling method can transfer 3D shape feature entities into 2D entities with relationships among these entities, which can be modelled by using object-oriented scheme structure.

### 4.4 The Information Integration Framework

The information integration framework aims to build an information bridge to fill the gap between the sheet metal part design, process planning, simulation and manufacturing systems. It is an important issue in supporting integrated and concurrent product development. This information integration framework mainly consists of a tree-based information modelling methodology to represent information objects and relationships. These objects represent the various information of a sheet metal product, such as parametric shape, topology and the design specifications (i.e. performance, materials, tolerance, surface and quality requirements), and manufacturing information (such as cost, manufacturability, process specifications, operational data). The relationships among different information objects (i.e. behaviour, state, etc) can also be modelled and can be used to build relational knowledge bases for supporting the concurrent product design and manufacturing. Also, the information framework can be used to build information models for information sharing among different software packages so that these software packages can concurrently work in an integrated environment.

#### 4.4.1 The Step Structure Information Framework

This research proposes an information framework called step structure information framework that is shown in Figure 4.5. The framework contains four top-down information layers, which include a knowledge layer, a parts layer, a feature layer and a parametric layer. The parametric layer contains the geometric data of the shape feature of the sheet metal parts and tool features. The feature layer contains all the feature information which includes not only the feature information (i.e. attributes) but also relationships with other feature-level information objects and objects defined by users. The part layer contains all the part information that includes feature information and relationships among different part-level information objects. The knowledge layer contains not only the part information, but also “knowledge related” information objects and an inference engine. The knowledge in the knowledge layer is extracted from part-level knowledge and feature-level knowledge, which are formed by information objects and relationships among them. The knowledge in the knowledge layer can be directly used to support intelligent concurrent design and manufacturing. The management feature is used to manage all the information of a certain part, which can be saved and used as a part of a company database. Application objects defined by users according to the requirements of the project can be put in the feature and part layer. After a new object is defined, its relationships will be created by either the users or automatically by the optimisation algorithm and existing knowledge. This object with its relationships can be regarded as a new knowledge, which can be used in new product design and manufacturing process.
In addition to geometry, this step structure framework that has been developed within this project includes representation of function and behaviour, physical decompositions, functional decompositions and the relationships between the physical and functional domain. In this step structure information framework, the relationships are used to represent the relationship between sets of objects, including a physical decomposition, a functional decomposition and other kinds of relationships. The overall sheet metal part relationships among shape feature representation is comprised not only of the collection of objects that represent physical entities, but also the other objects, relationships, the interconnections between them as well as various attributes and their values.

### 4.4.2 The Information Relationship Representation of Sheet Metal Parts

![Diagram of information relationship representation of sheet metal parts](image)

Figure 4.5 the information integration framework of sheet metal parts

From the discussion above, we know that one of the challenges in developing a sheet metal information framework is to delineate and represent the complex relationships among product entities and diverse types of sheet metal definition data. In conventional CAD/CAM systems, sheet metal parts are represented and stored as a complete geometric and topological solid in 3D. Such a representation is quite suitable for display and for performing geometric computation intensive tasks such as engineering analyses and simulation. However, it is not appropriate for tasks that require decision-making based on high-level information about geometric entities and their relationships. Therefore it is necessary to define sheet metal geometric entities and their relationships so as to provide high-level information required in the applications.

Tree-based feature models are able to represent the relationships among data and they can model sheet metal parts with the following results: (1) the representation of sheet metal geometric features, their properties, relationships and functions, (2) a wide definition of data and entity relations and (3) representation of semantics for different levels of applications. Knowledge can also be modelled by using this tree structure as it can be considered as the “efficient” combination of different information objects by an inference engine.

To represent the feature model and its relationships to the sheet metal part, the tree-based information models should not only represent the shape features, but also the relationships among different features. Usually, shape features contain both main features and auxiliary features. According to the relationships among main features, and between main features and auxiliary features, it is possible to build a relationship
graph among different features. The shape feature relationship graph can be used to represent the shape modelling process. Figure 4.6 is an example of a sheet metal part that will be used to show how to build a shape feature relationship graph. There are five planes contained in this part named \( P_1 \), \( P_2 \), ... \( P_5 \) respectively. There are several features (circle, rectangle, etc.) in each plane and there are four bend features named \( B_1 \), \( B_2 \), \( B_3 \), \( B_4 \) in this part. For this typical sheet metal part, the information object of each feature can be built according to the principles below.

The information modelling principle is as follows: it starts from a reference plane, and then joins other planes together based on this reference plane. When defining a bend feature, a reference plane feature that connects with this bend feature is chosen. Theoretically, from the information modelling point of view, the reference plane can be any plane in the part. But, usually one of the planes that will be fixture planes during the manufacturing processes will be selected as the reference plane. When defining a plane feature, besides the reference plane, it is necessary to choose another bend feature which shows how the plane contains this plane feature connects with its father plane. When defining an auxiliary feature, it is necessary to choose a plane feature from the same plane, then define this auxiliary feature according to the working coordinate of this plane. Figure 4.7 shows the tree-based relationship graph among different features for this example part when \( P_1 \) is supposed as the reference plane, the relationships among different features \((A_1, B_1)\) and planes \((P_i)\) are clearly shown in Figure 4.7.

Based on the principles above, I can use two branches of the tree structure to build any product models, which can make it easy to achieve different application modules, e.g. nesting optimisation, path planning and CAM module. The details of the two branches tree based information modelling methodology are as follows: The reference plane is regarded as the starting point of the tree. When defining a bend feature, it will be inserted into the left point of the tree or the left point of the left point of the tree, which contains the father feature that connects with this bend feature. This structure shows that the bend feature has a “son-father” relationship with its “father” feature; when defining a plane feature, it is being inserted into the right point of the bend feature that connects this plane with its father plane; when defining an auxiliary feature, it is being inserted into the right point of the tree or the right point of the right point of the tree, which contains a plane feature, which means the auxiliary features have brothers’ relationships with each other. Based on this modelling methodology, the example part can be represented as a two branch

---

Figure 4.6 an example for building feature relationships graph for a sheet metal part

Figure 4.7 the tree structure of shape features
tree structure which is shown in Figure 4.8.

It is obvious that this two branch tree-based data structure can also be used to describe the relationships among features, information objects and other user-defined objects. This tree structure can be extended as sheet metal parts become more complicated. However, building information models that clearly show the relationship among features in different planes is difficult and very important for the information framework. It should be mentioned that the relationships here contain static relationships and dynamic relationships. Static relationships mean that the relationships are not changeable (i.e. pre-defined material, tool-tool feature). Dynamic relationships mean that the relationships can be changed (i.e. a new punch tool object for a feature, an information object changes its connection to another object). Dynamic relationships usually are used to represent renewable knowledge and relationships among information objects. In an actual application process, the data structure that shows the relationships among different features may change as the information model goes through the application software, i.e. the relationships among different features in a CAPP information model will change according to the automatic arrangements of the shortest path optimisation algorithm. It is obvious that the tree structure can be well extended as sheet metal parts become more complicated and can also be used to represent distributed dynamic relationships.

In order to facilitate sheet metal product modelling and to build distributed information management systems, a common information model is built in this research. Figure 4.9 shows the structure of a common information model. It depicts a management object with its relationships to other downstream information objects, based on the information integration framework. The contents of information objects are defined by users based on the actual application (i.e. cost, tool sequence, lead time) and can be filled by software and users through the integrated system.

4.4.3 Modelling Languages and Methodology

The modelling language, that can be used to achieve the step structure information framework, should represent sheet metal parts as sets of objects and relationships of objects. It should represent physical entities such as assemblies, subassemblies, and components, as well as non-physical concepts such as function and behaviour in the Internet environment. In recent years, quite a few information modelling languages have been developed or are under development, e.g. the Integrated Computer Aided
Manufacturing (ICAM) Definition Language 1 Extended (IDEF1X) (Appleton D. company), the EXPRESS language (ISO 10303-11:1994E) and the Unified Modelling Language (UML) (UML Website). These information-modelling languages provide various ways of formally representing the information framework.

The modelling languages that have been used for this research include Visual C++, VBSCRIPT, JAVASCRIPT and EXPRESS. Visual C++ is an object-oriented (O-O) language, which supports class definition and object operations. It is used for developing most of the sheet metal application modules and tools. VBSCRIPT and JAVASCRIPT are used for knowledge sharing and client-server communication. EXPRESS is based on programming languages and O-O paradigm, and can be used for object definition and specification of constraints on the objects defined. A number of languages have contributed to EXPRESS, i.e. C++, SQL, etc. These languages make it easy to achieve all the application modules of the integrated system.

Figure 4.9 shows a management object and its relationships with other information objects that can be represented by EXPRESS or other O-O language. It includes a two-branches tree object for shape features, management feature object, material feature object, etc. The contents of an information object can be defined based on the actual application (i.e. cost, tool sequence, lead time, etc.). The shape feature objects of the part are stored in a two branch tree. The material feature object, the precision feature object and other objects are connected with the shape feature object. The relationship between precision feature object and shape feature object can be shown by the relationship between these two objects. The information object in Figure 4.9 can be extended as the two branches tree connects with more features and information objects.

4.5 Case Study

This section will discuss how to build an information model for a typical sheet metal part

Figure 4.10 shows a simple sheet metal part, which includes two plane features P_1, P_2 and one bend feature B_1. There are two inner rectangular features F_1 and F_2 in plane P_1. Suppose P_1 as the reference plane, and the feature F_1 is positioned by the reference plane P_1. The feature F_2 is positioned by the reference plane P_1 and the feature F_1’s position. The shapes and dimensional positions of these features have been shown on Figure 4.10.

Figure 4.11 shows the step structure information model of this part and the details of information models being built for different features of this part. The object-oriented expression of this part is shown in Figure 4.11 (a), which includes the relationships among different information objects, a management object and connection to other information objects. The relationships among different features, the relationship between feature and related information objects, etc. can be shown by EXPRESS-G. Other information models that represent other features are discussed as follows:
Chapter 4 An Information Framework for Rapid Sheet Metal Part/Product Development

The object-oriented expression of the plane feature is shown on Figure 4.11 (b), which includes information, such as the restraint value, the length and width of the plane feature. The object-oriented expression of the rectangular feature is shown in Figure 4.11 (C), which includes information, such as the restraint value, the length and width of the rectangle. The object-oriented expression of bend feature B1 is shown in Figure 4.11 (d), which includes information, i.e., bend radius and angle. The object-oriented expressions of two plane features P1 and P2 are similar; the object-oriented expressions of two rectangular features F1 and F2 are similar. The feature number of plane feature P1, bend feature B1 and plane feature P2 are 01, 02, and 03 respectively; the feature number of rectangle inner hole features F1 and F2 are 0101 and 0102. The whole information model of this part includes all these sub information models has been shown in Figure 4.11 (A). As we can see from this structure, the information model can be extended while the tree-based data structure has more classes or objects connected and can be divided into different layers. Also the detailed contents of the information model may change as the application changes, but the step information structure of this part would not change.

From above, we know that if we add a new feature into this part, a new information model will be created, this information model and its relationships with other features will be automatically added as they go through design and manufacturing systems. The data value of each object can also be changed by program automatically. For example, the value that means manufacturing sequences of the feature in a certain feature object will change as the information object goes through the shortest path optimisation algorithm. The cutting tool object has some relationship with these features. The tool planning module will assign each feature a relationship with a selected tool that will be used to manufacture this feature according to the optimisation results. The object view of the tools contains all the associated information elements (attributes). The information elements are, e.g., the tool ID, tool name, description, etc. Furthermore, there is also usage information, which will maintain the information relating to the level of inventory, tool life, tool location, etc. As we can see, the information schemas and relationships between features, tools, etc. can be implemented in the form of a company database.
4.6 Chapter Conclusion

In order to support the rapid development of sheet metal parts/product, an information framework that aims to provide information modelling methodology to support sheet metal parts intelligent concurrent design and manufacturing has been put forward and explained. This information framework divides information into four layers, which are a knowledge layer, a part layer, a feature layer and a parametric layer. It is an important step towards the development of information models of different parts that are responsive to the needs of emerging manufacturing and engineering technologies. This framework can also be used to achieve intelligent decision support in the earlier design stage through correctly defining the structure of the company's knowledge bases, which includes the representation, capturing, sharing and reuse of corporate design and manufacturing knowledge. The models extend well beyond representing solid geometry alone, and can, in principle, represent other important engineering information, i.e. properties, attributes and functions, and can be extended to the modelling process for other products.
CHAPTER 5

COMPOUND SHEET METAL CUTTING AND PUNCHING METHOD TO SPEED UP THE SHEET METAL PART/PRODUCT DEVELOPMENT

In sheet metal processing and manufacturing, there are a lot of small- and medium-sized job shops. These small- and medium-sized OKP manufacturing companies have been facing keen competitive pressure in the market. This pressure has forced these companies to make every effort to shorten product development lead-time, improve the production efficiency, approach high quality standards, but at the same time cut down the costs. To meet the needs of these companies, this chapter presents a compound cutting and punching production method supported by an integrated CAD/CAPP/CAM system in sheet metal manufacturing. Many existing commercial CAD/CAM systems are not suitable for this manufacturing method, especially under concurrent and global design and manufacturing environments. Some problems have to be solved before these CAD/CAM systems can be employed and integrated for this compound manufacturing method. This chapter deals mainly with the solutions to solve some of these problems. The solutions include an integrated data integration platform based on Pro/INTRALINK and STEP, and a knowledge-based real time CAPP (RTCAPP) system for compound sheet metal cutting and punching. Within the presented CAD/CAPP/CAM system, some key modules have been developed. They are the automatic tool selection and manufacturing sequencing module, a shortest tool path optimisation module, a cost estimation module and an automatic insertion of auxiliary path module based on knowledge bases. These modules will be addressed in this chapter.

5.1 Introduction

Sheet metal processing includes various manufacturing methods such as punching, blanking, bending, laser cutting, flame cutting, etc. Different manufacturing methods have different procedures, parameters and processes. Usually a sheet metal part is produced with different processes that may need to be manufactured on different machines, this multi-machinery requirement influences not only the manufacturing efficiency but also the manufacturing quality and cost.

The compound manufacturing method means two or more manufacturing processes can be carried out on one machine. The compound includes two common processes in sheet metal manufacturing, i.e. cutting and punching. By using this compound method, the cutting process and punching process can be carried out sequentially or concurrently in the same CNC sheet metal compound machine without altering the fixtures. This compound manufacturing method takes the high efficiency and low cost advantages of the CNC punching and makes use of the high flexibility of the CNC cutting for complex contour cutting.

Figure 5.1 a sample sheet metal part for compound manufacturing
According to the author's experience, this compound manufacturing method can be implemented economically. It is particularly suitable for those companies that produce sheet metal parts in a wide product range within short lead times. Figure 5.1 shows an example that needs to be processed through the compound manufacturing method, since the special-shaped parts 1 can only be cut whereas all the other parts can be directly punched.

An integrated CAD/CAPP/CAM system is very important for these sheet metal processing companies to support the implementation of the compound manufacturing method. Nowadays CAD/CAM systems that will enable a small or middle-scale company to produce high quality products in a cost-efficient way are much needed. Many existing commercial CAD/CAM systems are either too expensive or too narrow in scope, in the sense that they provide only point solutions (Chaturvedi and Allada 1999). The design of parts and the planning of their manufacturing processes are highly specialized and knowledge-intensive in nature (Sakal and Chow 1994). The user is forced to use two or more CAD/CAM systems that must be compatible with each other, while many of these systems are stand alone and expensive for small or medium-sized companies. The integrated CAD/CAPP/CAM system should incorporate specific design and manufacturing methods in the companies.

Several integrated CAD/CAPP/CAM systems have been reported in the sheet metal manufacturing area. Wang and Bourne (1995) presented an automatic process planning for bent sheet metal parts. Ismail et al. (1993) developed a low-cost integrated computer-aided press tool design (CAPTD) system, which uses constraint-driven and object-oriented techniques to advise users on the design of press tools. This CAPTD system uses AutoCAD as the design platform, dBase IV is used for the selection of press tool components and a C program based routine is used for the optimum layout of the work-piece. Lee et al. (1993) have developed a knowledge-based process planning system (IKOOPP system) to automate and standardize the process planning function for the manufacture of progressive dies. The IKOOPP system receives the part definition data from the commercial die design system. The relevant attributes of the manufacturing features within a progressive die model are extracted by a feature extractor and then used by the IKOOPP system to generate the process plan by using knowledge bases. Lauwers and Kruth (1994) developed a computer-aided process planning system for an electrical discharge machining (EDM) process that can generate an optimal process plan based on the minimal cost criteria. An integrated intelligent process planning system for prismatic parts was reported by Sakal and Chow (1994), their system integrated AutoCAD and MasterCAM software packages using Autolisp programming language and artificial intelligence (AI) techniques. Park (1993) presented a real time CAPP system in order to achieve concurrent design of prismatic parts and their manufacturing processes.

Despite the success of the systems described above, there exist some limitations to their application when extending them to compound manufacturing processes. These systems need to be further developed to overcome some key problems so that they can support the compound manufacturing processes, especially for compound cutting and punching processes. These problems include: automatic manufacturing process sequences generation, the mathematical model and optimal algorithm for tool selection and manufacturing path optimisation, nesting heuristics and the automatic insertion of "suitable" auxiliary paths for cutting processes based on knowledge bases. Also, all these functions have to be considered in a timely manner in order to support concurrent design and manufacturing. To solve these problems, an integrated CAD/CAPP/CAM system has been developed by the author and it will be described in this chapter through the discussions made on its key components as follows.

1) A data integration platform. This platform is developed based on Pro/INTRALINK database structure and its API. This platform, which is for the definition of application data and functions, aims to cover all data forms of the whole lifecycle of sheet metal parts, i.e. from their design to manufacturing. The data integration in this platform is based on the STEP standard and object-oriented modelling methods. The platform involves multiple design and analysis applications that require data from several STEP Application Protocols (AP203, 209, 214, 224) and integration resources from different parts of STEP. Object-oriented modelling methods are used for data that are beyond the STEP standard, i.e. aerodynamic analysis, dynamic
data, parametric geometry and constraints. For information that is beyond the STEP standard, “extended” objects are created based on STEP and an object-oriented modelling method. The information-modelling framework, as addressed in Chapter 4, is used to build information models for CAD/CAPP/CAM integration. Based on this framework, information being transferred among different modules in the CAD/CAPP/CAM integrated system can be represented using any object-oriented language. This approach has also been used in a computer-aided system for concurrently designing and manufacturing a customised OKP mould or die that has complicated sculptured surfaces (Tu et al. 1998). The platform provides a data integration environment for this integrated CAD/CAPP/CAM system.

2) A knowledge based RTCAPP module. This module includes several sub-modules, i.e. automatic insertion of auxiliary paths based on knowledge bases, a nesting module and a sub-module for tool sequencing and tool path planning.

3) A simulation platform has also been developed for this integrated CAD/CAPP/CAM system. In this module, an example will be given to compare the compound manufacturing method based on this integrated CAD/CAPP/CAM system with traditional single manufacturing methods.

5.2. Integrated System Architecture for Punching and Cutting

Because of industries increasing needs for other types of information or knowledge in sheet metal design and manufacturing process, new classes of tools to support knowledge-based design, product data management and concurrent engineering have begun emerging in the engineering market. However,

![Diagram](image)

Figure 5.2 the CAD/CAPP/CAM integrated system structure for sheet metal compound manufacturing

traditional CAD, CAM systems usually run independently and the information transfer among these systems is through certain file format (STEP, DXF, etc.). Also, traditional CAD, CAM tools are mainly focused on database-related issues and do not place a primary emphasis on information models for product representation (Szykman et al. 2000). A favourable integration environment is critical to overcome these disadvantages. Most applications are inclined to use commercial software packages and secondary development tools attached to them to establish the needed integration environment.

Pro/Engineer series from PTC is an integrated software package. One of its core technologies is parametric and feature-driven modelling and the other is that design and manufacturing share a common related database by using Pro/INTRALINK. The common space database (the shared database within the Pro/INTRALINK environment) and object-oriented API can be used as a tool to develop an integrated data-sharing and communication environment, where complete information integration overlapping the whole life cycle of a product can be achieved. The common space database (the shared database within the
Pro/INTRALINK) is used as a collection point of design activities. The data in this environment can not only be shared by different commercial CAD and CAM platforms through file transfer, but also be accessed concurrently through Internet and Intranet communication. The CAD/CAPP/CAM integrated system for sheet metal compound manufacturing is developed based on Pro/Engineer platform.

The overall structure of the sheet metal compound manufacturing CAD/CAPP/CAM integrated system is shown in Figure 5.2. There are several modules in this system, including an unfolding module, a data integration platform, a knowledge based RTCAPP module, a CAM module, a cost estimation module and a simulation module. The unfolding module in this system adopts zero thickness and zero bending radii principles in order to facilitate sheet metal part design processes. The data integration platform is a data integration environment for data exchange and sharing between CAD/CAPP and CAPP/CAM that is developed based on Pro/INTRALINK and API. The knowledge based RTCAPP module generates a process plan, manufacturing parameters, tool sequences, cost, etc. and sends all the information back to the data integration platform in real time so that other modules can use them. The simulation module is designed to test all the modules. This chapter will discuss three major modules; the data integration platform based on STEP for sheet metal compound manufacturing, the knowledge based RTCAPP module and the simulation module.

The knowledge based RTCAPP module for sheet metal compound manufacturing contains several sub modules, including knowledge based auxiliary path automatic insertion, an optimisation module, an experimental database and a tool and module library. The optimisation module is the most important module in order to achieve efficient and economic manufacturing and includes tool sequencing, tool path optimisation and nesting optimisation of sheet metal parts with complicated shapes, etc.

There are also some other situations that need to be dealt with in this RTCAPP system, e.g. the selection of a suitable machining method (cutting or punching in this project). To produce a contour of a sheet metal part depends on many factors, for example, the degree of complexity of the contour, the batch size of the parts, machining precision, the thickness and material of the sheet metal part. To determine the sequence of the processes, the system must make sure that the outer contours are machined last so as to avoid machining parts that are not fixtured. Tool sequence planning is also important for both the compound manufacturing process. Tools are often changed according to different features of a part or product, since different tools are usually needed for producing different features. The frequent tool changes in turn lead to lower production efficiency. Hence in order to save the tool changing time, tool sequence planning is needed. Tool path planning is also very important to improve the efficiency through selecting shortest machining paths and hence shortening non-machining time. Knowledge bases are very important for the determination of cutting parameters, lengths and shapes of the auxiliary path and manufacturing processes. In the sheet metal laser cutting process, for instance, part materials, pressure of auxiliary gas and laser power will influence cutting speed and acceleration. Some major modules will be further discussed in the following sections.

5.3 Data Integration Platform for Sheet Metal Compound Manufacturing

For CAD/CAPP/CAM integrated systems, the data integration is very important from the standpoint of more effective utilization of data in engineering enterprises based on computer-aided systems (Chen et al. 1994). Complete information integration overlapping the whole life cycle of a sheet metal product is the basis of data sharing and communication among CAD, CAPP and CAM sub systems. From Figure 5.2, we know that a convenient and common data integration platform is essential for sheet metal compound manufacturing, as there is more possible data communication and exchange among different modules.

The data integration platform is based on the step-structure information modelling framework. This framework is based on STEP and the object-oriented method. STEP is used to represent geometry information and information requirements for a particular area of design, manufacturing, engineering or
product support (Duan et al. 1996). The framework contains four top-down information layers, which include the knowledge layer, the parts layer, the feature layer, and the parametric layer. Information models can be built based on this information framework for data exchange and sharing between CAD, CAPP, and CAM system. The parametric layer contains the geometric data of the shape feature of the sheet metal parts and tool features. The feature layer contains all the feature information which includes not only the feature information (i.e. attributes, constraints) but also relationships with other feature-level information objects and objects defined by users. The part layer contains all the part information that includes feature information and relationships among different part-level information objects. The knowledge layer contains not only the part information, but also "knowledge related" information objects and an inference engine. Application objects defined by users according to the requirements of the project can be put in the feature and part layer. The detail of this framework has been well addressed by (Tu et al. 2001) and has been used for information modelling of a concurrent manufacturing of mould/die products with complicated sculptured surfaces (Tu et al. 1998).

Figure 5.3 shows an extended common object-oriented data class structure based on the information modelling framework, the relationships with other classes or objects have also been defined on this platform. Other features can also be defined based on the common class. For example, a new shape-feature, which contains information that is beyond STEP, can be defined. Figure 5.4 shows an example of using object-oriented modelling and STEP to build an information model for an inner feature and a related punching tool. Classes and relationships between the inner feature and the tool are clearly shown in Figure 5.3. For punch tool selection, each inner feature, outer feature and non-feature element of the outer contour has a related tool after a tool has been automatically selected. An information model that shows the feature-tool chain data structure and parametric element-tool data chain structure is also shown on Figure 5.4. Other information models that are needed for this CAD/CAPP/CAM system can be built based on the framework (Tu et al. 2001).

The data integration platform based on the framework for sheet metal compound manufacturing includes two integrated graphical user interfaces. One is a CAD/CAPP data interface, which includes feature data processing, feature data recognition/extraction and feature data transformation. The geometry information from CAD is saved in the parametric layer. The CAPP information (i.e. cost, attributes, constraints, machining sequences, collision or
machining time) from CAPP to CAD is defined through the data interface and saved in the feature layer (see Figure 5.3). The CAPP module can read the CAD information through accessing information in the parametric layer. Also, the CAPP module can add process information related to CAD features, i.e., auxiliary paths for features that would be cut will be added automatically, relationships will be created to show the tools that would be used to punch the feature. The other is CAPP/CAM data interface. The main task to integrate CAPP and CAM under Pro/ENGINEER is to receive geometry and partial process information from CAD and machining parameters from CAPP. In this compound machine (laser cutting and punch), the information exchange between CAM and CAPP include laser power, machine speed, laser focus position, punching tool information, and other machining parameters. The information flow between CAPP and CAM is different in different systems and applications, but there is no standard protocol between CAPP and CAM that can be abided by (Papadimitriou et al. 1987). Thus, these two interfaces are designed to solve this problem. They can be used for non-standard protocol data and functions definition, such as cost information, specific manufacturing parameters and functions for specific manufacturing processes.

The data integration platform makes it possible for CAD, CAPP and CAM software to directly communicate with this platform concurrently. Other specific applications can also be carried out based on this platform. Real time process planning (Park 1993) can also be achieved based on this data integration platform if suitable information can be sent to the platform in real time.

5.4 Knowledge based RTCAPP system for CNC punching and cutting

Computer Aided Process Planning is one of the most important and complicated CA systems,

![Diagram of data structure and functions of integrated CAPP system for sheet metal compound manufacturing](image)

Figure 5.5 the data structure and functions of integrated CAPP system for sheet metal compound manufacturing
especially for sheet metal compound manufacturing. In recent years, advanced computer software
techniques such as expert systems and artificial intelligence approaches have been used to develop
generative process planning systems. The CMPP (Duda et al. 1997) and HZ-RCAP (Cai 1995) systems
apparently have a generative process planning functionality for cylindrical-shape parts, where knowledge of
manufacturing processes and parameters are in the rule base. The RTCAPP in this chapter presents a
knowledge based RTCAPP system based on the data integration platform, which includes several
optimisation modules that aim to improve the manufacturing efficiency and cut down costs and knowledge
based auxiliary paths and manufacturing parameter selection for sheet metal cutting processes. Several
typical functions of the CAPP system have been defined in Figure 5.5. The functions in the RTCAPP system
are designed to achieve different application tasks. These functions can access all required information
through the data integration platform. Tool database, cutting parameter database and process rule database
are all connected with the data integrated platform through ODBC. In Figure 5.5, functions have been
defined for automatic tool selection, tool path optimisation, nesting optimisation, manufacturing parameters,
etc. For the sheet metal compound manufacturing RTCAPP system, real time optimisation modules are
very important for improving manufacturing efficiency and cutting down costs, while manufacturing
parameters and auxiliary path will influence the manufacturing quality of sheet metal parts. Details of these
modules will be further addressed.

5.4.1 Tools Sequencing Real Time Optimisation Algorithm for Punching

The main effort in generating a production plan for punching operations lies in the process of tool
selection. In the modern sheet metal industry the selection of suitable tools is a major economic issue for
two reasons. Firstly, tooling is very expensive and should be minimised. Secondly, if the tool capacity of the
tool rack is exceeded, expensive set-ups are needed, so tool sequencing and arrangement are very important.
On the punching process of sheet metal parts, if the tool is to be changed according to different features,
different tools are needed for different features, so tool changing is needed frequently and will lead to lower
efficiency. In order to save tool-changing time, it is necessary to finish all related feature-machining
procedures after choosing a tool, and then to choose another tool until all machining procedures finish.

A tool-sequencing algorithm that is used to arrange the tool sequences is shown as follows:

1. According to the feature-tool chain-list structure, find out all the tools which can manufacture all
the inner and outer features for this part, and construct a tool-tool position chain-list, as shown by
Figure 5.6;

2. From tools-tools position chain-list structure, search
a tool position \( P_i \) nearest to the origin of a working
coordinate;

3. Find related tool
   \( T_i \) according to tool
   position \( P_i \);

4. Tool path planning for all
   tool \( T_i \) related tool
   position points, this algorithm will be given on 5.4.1.2;

Figure 5.6 Tools-Tools position chain-list

-66-
(5) Decide if all the tool machining sequences have been arranged, if finished, then algorithm ends, otherwise set the last tool position as \( P_E \), then search the tool position \( P \) nearest to \( P_E \) from the tool positions on which the related tool sequence was not arranged, set \( P = P_E \), to (3).

With regard to a single punching tool, the arrangement of the punching sequence is to ensure that the total distance that the punching tool moves is the shortest. It is a tool path optimisation problem. If a machining path can be arranged reasonably, then tool non-machining time can be cut down.

5.4.1.1 Mathematical Model

The punching tool path planning question can be described as: 1) A tool begins to punch from the current position; 2) It should finish all related punching operations, and the punching operations can not repeat on every certain position; 3) An optimisation algorithm needs to be developed to automatically select a manufacturing sequence by which the tool's moving distance is the shortest.

The tool path optimisation problem can be transferred to a mathematical problem for the shortest Hamilton return-loop in diagram theory. For convenience, the start position and all the related position description for all punching operation of the tool can be described as a point set \( P : \{ p_1, p_2, \ldots, p_n \} \). Hence, the tool path optimisation problem can be described by using diagram theory as follows:

Construct a whole diagram \( K_e = (V, E) \), among them, \( V \) is a set of top points and \( E \) means the distance between each point. From all top points, the algorithm seeks a return-loop passing through all points once and just once, to make sure the sum of distances that this tool moves is the shortest. This is a Hamilton Loop optimal problem.

In fact, this is an NP-completeness problem. Some algorithms could be used for this problem under different situations, such as dynamic programming, tree theory, etc. There are no efficient algorithms that can find the exact solution within polynomial time (Summad and Appleton 1998, Papadimitriou and Steiglitz 1987, Hamel and Steel 1994). A reasonable method is to find an approximate algorithm to solve this problem. In our system, the shortest insert algorithm has been adopted, which can be easily implemented and it meets the concurrent requirement of design and manufacturing.

5.4.1.2 Mathematical Description of The Shortest Insert Algorithm

The distance between a sub return-loop \( T \) and a top point \( P \) which does not belong to the return-loop is defined as:

\[
\text{d}(T, p) = \min \{\text{d}(x, p) : x \in T\}.
\]

Suppose the whole graph is expressed as \( K_e = (V, E) \), and the number of points in this graph is \( n \). Then the mathematical description of the shortest insert algorithm is given below:

a. Choose a top point \( v_1 \), and a sub return-loop \( T_1 \) which is formed without side. Then find a top point \( v_2 \) which has shortest distance from \( v_1 \). Based on \( v_1 \) and \( v_2 \), another return-loop \( T_2 \) is formed, which includes sides: \( (v_1, v_2) \) and \( (v_2, v_i) \):

b. If sub return-loop contains all \( n \) top points, then algorithm ends, otherwise, seek a top point \( v_i \in V \setminus T_1 \), this point meets the condition \( \text{d}(T_1, v_i) = \min \{\text{d}(T_1, x) : x \in V \setminus T_1\} \), \( 1 < i < n \);

c. Find side \((v_x, v_y)\) from \( T_1 \), which makes that \( \text{d}(v_x, v_y) + \text{d}(v_x, v_y) - \text{d}(v_x, v_y) \) is the shortest distance. Delete side \((v_x, v_y)\) from \( T_1 \), a new return-loop \( T_{i+1} \) formed by sides \((v_x, v_y)\) and \((v_x, v_y)\):
d. The top point number in this sub return-loop plus 1, then go to step b.

5.4.1.3 Programming Implementation

The steps for implementing the nearest insert algorithm are described in the following:

a. Build a matrix \([d_{ij}]\), which shows distances between different top points.

b. Create a linear list: "List1", and put all the top points in List1. Build a loop chain-list: "List2", put the first point A in List1 into List2, and delete this point from List1.

c. Calculate the distances between every point in List1 and the point A in List2. According to the calculation results, find the point B, which has the shortest distance. Insert B into List2 and delete point B from List1.

d. If List1 is NULL, then algorithm ends, otherwise compare the distances between all points in List1 and all points in List2, and find the two points with shortest distance (suppose the point in List1 is x).

e. For all neighbouring points \(M_i\) and \(N_i\) from List2, calculate the absolute value of \(x \cdot M_i + x \cdot N_i - M_i \cdot N_i\). Select the two neighbouring points \(M\) and \(N\) with the smallest absolute value.

f. Insert x in the middle of \(M\) and \(N\) on List2, and delete x from List1. Go to (d).

After the algorithm finishes, linear chain List1 is empty, and all top points have been stored in return-loop list List2. The tool sequence and related position have been put in List2, and need to be put in related tool objects, a correct connection is automatically created between the tool object and related features that would be punched by this tool. After the tool-sequencing algorithm finishes, each feature that would be punched should have been connected to a tool object.

5.4.1.4 The Performance of the algorithm

The performance of the shortest insert algorithm is very important in the CAPP system. Several tests have been carried out with regard to its speed, optimality and robustness. Table 5.1 shows three major characteristics of this algorithm compared with two other common algorithms. In this table, \(L_a\) is the length of the Hamilton return-loop and \(L_b\) is the optimum length of the Hamilton return-loop. As we can see, the nearest neighbour algorithm is easy to implement by programming, but its optimum performance is not good. The optimum weight spanning-trees algorithm can achieve better optimisation results, but it is hard to achieve using programming and takes a long time to find the optimum result, especially when sheet metal parts become more complicated.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time complex</th>
<th>Algorithm performance ((L_u/L_b))</th>
<th>Programming implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest neighbour algorithm</td>
<td>(O(n^2))</td>
<td>(\leq \frac{1}{2} (\log_2 n + 1))</td>
<td>Very simple</td>
</tr>
<tr>
<td>Shortest insert algorithm</td>
<td>(O(n^2))</td>
<td>&lt;2</td>
<td>Simple</td>
</tr>
<tr>
<td>Optimum weight spanning-trees algorithm</td>
<td>(O(n^3))</td>
<td>&lt;2/3</td>
<td>Most complicated</td>
</tr>
</tbody>
</table>

Table 5.1

As discussed in 5.4.1.3, the shortest algorithm can be easily implemented by programming and performance is better than the nearest neighbour algorithm and is not time consuming (from table 1), which is suitable for the real time optimisation task in this project. In order to test the actual application results of this algorithm, a series of tests have been carried out in two sheet metal companies. A case study will be given in section 4 (see Figure 5.10). These tests show that the algorithm can achieve good optimisation results and has high robustness and stability.

5.4.2 Knowledge Based CAPP for Sheet Metal Cutting

Manufacturing parameters and auxiliary cutting path are very important for the cutting quality of sheet metal parts if a cutting procedure is chosen. The automatic selection of manufacturing parameters by using experiment results and neural network reasoning method was well addressed by several authors (Raggenbass and Reissner 1991). Auxiliary path is another difficult problem for sheet metal cutting, especially for flame or laser cutting. For sheet metal parts with different shapes, the auxiliary cutting path is quite different. Usually the auxiliary cutting path includes three types: “cut-in”, “cut-out” and closed auxiliary path.

Usually, the cutting process does not start from the contour directly, but from somewhere else. A hole should be made (using laser or flame) first, and after cutting a segment of auxiliary path, the cutting of the contour should begin. After contour cutting is finished, another exit auxiliary path is needed to be cut; these two segments of auxiliary path are called “cut-in” and “cut-out” auxiliary path. The hole is called “start hole” which should be set on waste material. When cutting inner contours, this hole should be set on the inner side of the contour, while cutting outer contours, the hole should be set on the outside of the contour. Closed auxiliary path is a special auxiliary path that should be added in order to avoid burning the sharp angle while cutting it as the outer contours. There are several different rules for different features, some examples are given below. All these rules have already been put into the knowledge base.

5.4.2.1 “Cut-in” and “cut-out” auxiliary path

(1) How to choose “cut-in” point

The “cut-in” point is the first point in a contour after cutting the auxiliary path. This point is set automatically by program. The CAPP platform also provides a user interface for setting this point. The automatic generation of the “cut-in” point is based on the principles as follows:

a. If the inner contour or the outer contour is circular, choose the point with least y coordinate as the “cut-in” point;

b. If the outer contour is organised by line segment and circular segment, a set of top points, U,
will be formed by neighbouring parametric elements. Suppose \( U_x \) is a set of the points with least x coordinate from set U. From \( U_x \) the point P with least y coordinate as the outer contour “cut-in” point can be chosen;

c. If the inner contour is composed by line and circle segments, find top point P according to step b. From all the inner contours, find a related line or circle segment the end point of which is P, and then choose the middle point as the “cut-in” point.

(2) Determination of “cut-in” and “cut-out” auxiliary path

In fact, “cut-in” or “cut-out” auxiliary path is a segment of circle or line. We also call them “cut-in” “cut-out” stretching arc. Generally, for inner contour, “cut-in” or “cut-out” stretching arc is a segment of circle. But for outer contour, generally the “cut-in” or “cut-out” stretching arc is a segment of line.

As shown on Figure 5.7, the “cut-in” and “cut-out” stretching arcs of outer contour are the extending lines of the tangent line of outer contour at the “cut-in” point. On Figure 5.7, lines from point A to point P are “cut-in” stretching arc, line from point P to point B are “cut-out” stretching arc.

![Figure 5.7 “cut-in” and “cut-out” arc for outer contour cutting](image)

As shown on Figure 5.8, “cut-in” and “cut-out” arcs of inner contours are the circular segments, the directions are tangential to original line or circle in point P, on Figure 5.8, lines from point A to point P are “cut-in” stretching arc. Line from point P to point B is “cut-out” stretching arc.

![Figure 5.8 “cut-out” and “cut-in” arc for inner contour cutting](image)

While cutting the sharp angle of outer contours, in order to avoid burning the sharp angle, a circular auxiliary path is needed. A closed auxiliary path can be added on the convex point of outer contours. Adding an auxiliary path on concave point is not allowed because it will “burn” the outer contours when the cutting
tool enters from auxiliary path to contour. There are three kinds of sharp angle situations shown on Figure 5.9: Line-Line, Line-Circle and Circle-Circle, the added closed auxiliary paths as shown on Figure 5.9 are small triangles, the direction of the entry and exit paths of the auxiliary path are tangential to the original contour paths shifting at the corner.

5.4.2.2 least distance between “start hole” and contour

The “start hole” is a small hole firstly made before cutting the outer contours. A certain amount of time is needed for making this small hole. Hence, there is an area called the heating influence area. In order to enhance manufacturing quality, the heating influence area cannot extend to the contours of part. This means a least distance between “start hole” and contour must exist. How to choose this least distance is influenced by cutting pattern, part material and thickness, and so on. To automatically choose this least distance, a database was built based on experiments in factory and a neural network aggregated from these experiments was also developed in this CAD/CAPP/CAM system.

The knowledge base that contains how to insert the “cut-in” and “cut-out” auxiliary path is very important to the manufacturing quality of cutting processes. After the knowledge base was built according to the rules above, a suitable auxiliary path can be inserted automatically according to this knowledge base and experiment databases. The experiment databases were built as a result of two years of cutting experiments in the factory; the reasoning module by using a neural network with a self-learning function has been used in this system, which can automate the cutting process planning. This module is very important for compound manufacturing processes, without this module, the RTCAPP for compound sheet metal parts cannot be achieved.

5.4.3 Tool Path Optimisation for Cutting and Cost Estimation

The cutting tool path optimisation is also important in order to improve cutting efficiency. Compared with the shortest tool path optimisation algorithm for the punching process discussed in 5.4.1, the cutting process for sheet metal parts is different from CNC punching and tool exchange is not needed. The principle of the shortest path optimisation algorithm for both cut and punch is almost the same, while some restrictions need to be considered while implementing the optimisation process for cutting. These restrictions are based on the actual compound manufacturing requirements. The cutting and punching sequences need to follow some rules while the optimisation processes are considered, i.e., both punching and cutting processes must obey the rule whereby the inner contour must be cut or punched first and outer contour cut or punch later. Hence, the tool path optimisation for CNC cutting includes not only the path planning for the inner contours of each part, but also the machining sequence of parts after nesting. Generally, for compound manufacturing of cutting and punching, nesting optimisation of the different shape of parts should start first. The tool selection and punching path-planning processes have to wait until the nesting optimisation process finishes. The cutting process does not manufacture every part separately but depends on the shortest cutting distance optimisation algorithm. Regarding the cutting sequence of each part,
the entry point P of the part can be used for the arrangement of manufacturing sequences. For tool path planning of the inner contours of each part, the entry point P of the inner contours can be used for arrangements of machining sequence; the cutting path optimisation algorithm is the same as the punching tool sequencing algorithm, which has been discussed in 5.4.1.2.

The manufacturing cost estimation sub-module estimates the manufacturing cost of the sheet metal part. This sub module takes the input from the NC programming and the knowledge based RTCAPP sub module. The compound machining time is calculated from the NC part programs. The set-up times are estimated times associated with a particular punching operation, which include punching times and set-up times. All the cost information has been recorded in a database. The cost of using cutting processes such as the expense of cutting gas, auxiliary gas, cutting machine and laser machine for laser cutting can be calculated by machining times. The cost of product design and process planning can be also calculated by related cost information in the database. The cost of sheet metal utilization ratio can be calculated according to the data input from the nesting optimisation algorithms sub module. The total manufacturing cost of the part can be calculated by adding the estimated material cost to the sum of the costs of all processes. As we can see, all the optimisation algorithms have great influences on cost. The function of cost estimation has been defined in Figure 5.5, which can be sent to CAD or other modules.

5.5 Simulation Platform for Compound Manufacturing

Simulation is one of the most powerful analysis tools available for the design and operation of complex processes in a distributed environment. For compound sheet metal cutting and punching, a simulation platform becomes more important because of its complex manufacturing process. Which both punching and cutting processes run concurrently. In this CAD/CAPP/CAM integrated system, a real time simulation platform has been developed. This simulation platform is directly driven by geometric parameters, manufacturing parameters, and process planning results. All key components have to be tested in this platform, such as punching tool selection, path optimisation, nesting optimisation, etc. Some modules have been developed and tested alone in the companies for about two years before integrating to this integrated CAD/CAPP/CAM system, such as nesting and shortest insert algorithm. A case study has been given with regard to the tool path planning, the performance of the optimum algorithms (nesting and shortest insert algorithm) and auxiliary cutting paths in two sheet metal companies. The single and compound manufacturing processes have been compared.

Figure 5.10 shows the different manufacturing processes of four frame parts after nesting optimisation by using different manufacturing methods. As there was no nesting algorithm in these companies, the nesting program greatly improved the utilization ratio of their raw materials, especially when the shape of the sheet metal part became more complicated. Auxiliary cutting paths are added automatically according to knowledge bases that are discussed in section 5.4.2 and which can be clearly seen from Figure 5.10 (1) and Figure 5.10 (2). In Figure 5.10, (1) shows the performance of the laser cutting method with the shortest path optimisation algorithm and (2) shows the performance of the laser cutting method with the nearest neighbour optimisation algorithm. For the example parts, the cutting distance difference between the two algorithms is shown on (8). It is obvious that the shortest path optimisation algorithm can reduce the non-machining moving distance, its performance is better than the nearest neighbour optimisation algorithm. Figure 5.10 (4) shows the punching machining method with the shortest path optimisation and Figure 5.10 (5) shows the punching machining without optimisation algorithm. We can see the difference before and after the tool path-planning algorithm. Comparing Figure 5.10 (4) with Figure 5.10 (5) the total path length after optimisation is even longer than the total path length before optimisation. However, punching tools need to be frequently changed according to Figure 5.10 (4), which wastes a lot of time. Figure 5.10 (7) shows the machining time difference of the single punching machining method before and after optimisation when tool-changing time is set as 50s and the moving speed of the machine is 5 m/s. The machining time difference is also shown in Figure 5.10 (7) comparing the single punching method and the compound method. The performance of the compound manufacturing method can be clearly seen by comparing with single manufacturing methods. Figure 5.10 (3) shows the compound manufacturing method that contains all
the functional modules. It is apparent that the total distance that manufacturing tools move are the shortest compared to other methods in Figure 5.10. This compound manufacturing method can greatly shorten the manufacturing time and distances that tools move. Manufacturing efficiency has been greatly improved. After being given related cost parameters mentioned in section 5.4.3, cost can be automatically estimated. Figure 5.10 (6) shows the cost estimation of the compound manufacturing method compared with two other single manufacturing (cutting and punching) processes before and after optimisation. From the simulation results above, manufacturing time and cost can be greatly cut by using the CAD/CAPP/CAM integrated compound manufacturing method. The utilisation ratio of sheet metal part can be improved by using the nesting algorithm in real time, costs can be optimised and can be sent back to the designer in real time and a more economical design can be achieved.
5.6 Conclusions of the Chapter

A fully integrated and concurrent sheet metal compound manufacturing CAD/CAPP/CAM system, which allows for the creation of a real time process plan and definition of non-standard protocol information for sheet metal part design and compound-manufacturing processes has been presented in this chapter. The source code of the CAD/CAPP/CAM system can be referred to the Appendix IV. A data integration platform
has been put forward for the definition of information models and functions for sheet metal punching and cutting. A knowledge-based real time CAPP system with optimisation of manufacturing process and utilisation ratio has been developed in order to improve manufacturing efficiency, product quality and cut down costs. This integrated and concurrent CAD/CAPP/CAM system eliminates the necessity of employing different software packages for simulation, NC generation and process planning and optimisation activities. It will enable "right first time" design, improve product quality, reduce product costs, and enable a rapid response to market needs in terms of reaction time and product changes. This integrated CAD/CAPP/CAM system for compound sheet metal parts manufacturing has been developed by using Visual C++ programming language. After the system was developed, it has been used by two sheet metal manufacturing companies for about two years. The utilization of their sheet metal parts has been improved by 15%, and costs, product design and manufacturing time have been greatly reduced, and the product development cycle time has been greatly shortened.
CHAPTER 6

AN INTERNET BASED COST ESTIMATION AND OPTIMISATION FRAMEWORK FOR RAPID SHEET METAL PART/PRODUCT DEVELOPMENT

In the early stages of a PD process, the estimate of a developing product life cycle cost is the primary information that will be needed to determine the profitability of the product. The more reliable the cost estimate process is, the more likely the right decision will be made. This has become an important issue for RPD. In this chapter, an Internet based cost estimation and optimisation system, which is a part or a sub-system of the IRPD system will be reported. The PD cost under consideration in this chapter is the sum of the costs of product design, production and logistics, which is the main cost of a product. Some other possible costs associated with a PD cycle, e.g. overhead costs, are normally proportional to this main PD cost and in practice they are estimated by multiplying a constant percentage to the PD cost. Unlike some earlier research which focused on the cost estimate in a single manufacturing company, this Internet based cost estimation system aims at the PD cost estimate and optimal control under the complexities of a global manufacturing environment. The proposed Cost Estimate and Optimal Control system has been implemented in a sheet metal manufacturing company. This industrial implementation will also be reported in this chapter to demonstrate the principle of the system.

6.1 Introduction

In today’s keen competitive global market, a short PD cycle time and high quality product with a reasonable lower cost are becoming more and more critical for these small and middle sized manufacturing companies to survive. In particular, the PD cost normally determines the market share, profit and return on investment of these small and middle-sized manufacturing companies. To control a PD cost at earlier PD stages is in turn a problem in how to accurately estimate a possible cost associated with a PD cycle. Estimate of PD cost is closely related to the selection of a mix of production development processes, which can fully meet the customer’s requirements and with a lowest sum of the costs of these selected processes. Hence, it is necessary to develop new methods and tools for these small and middle sized manufacturing companies to rationally select proper PD processes, accurately estimate and optimally control the PD cost.

Cost estimation and reduction have been always of great concern in a PD cycle. In recent years, the research which aims to improve a manufacturing company’s PD ability has been focused on two aspects, 1) how to shorten the PD cycle time, and 2) how to reduce the PD cost. In fact, these two aspects are interrelated to each other. A shorter product cycle time will normally result in a lower PD cost. On the other hand, to optimally control a PD cost usually leads to a shorter PD cycle time. Hence, the research projects, either study of the PD cycle time or study of the PD cost, aim to achieve the same goal, i.e. rapid and economic PD, just from two different angles.

To estimate and optimally control a PD cost in today’s global competitive manufacturing market, several issues need to be researched. First, PD in a global manufacturing environment has been proved to have the advantages of agility, often providing quick and cheap solutions for some parts or even the whole product development from the partners or sub-contractors, and maximally and yet effectively applying state-of-the-art technology to address the customer’s requirement in a global competitive environment. However, a well-established PD management system (e.g. cost estimate and optimal control system) in a global manufacturing environment is still under development. Although a relatively large number of papers
and reports were found to address the cost estimate and control problems in various PD processes, such as design, manufacturing and logistics chain management, these methodologies were normally developed based on the cases in individual companies rather than in a global manufacturing environment. In a global manufacturing environment, a manufacturing company (normally called master company) may carry out a PD cycle through collaboration among its partners or sub-contractors. These partners and sub-contractors may use different PD management systems and computer aided engineering software systems. To effectively communicate, share the necessary information, integrate the efforts and optimally control the PD cost between the master company and its partners/sub-contractors are the problems which need to be addressed by this research which aims to develop a PD cost estimate and optimal control system in a global manufacturing environment.

Second, no matter whether in a global manufacturing environment or in an individual manufacturing company, PD cost estimate and control is an interdependent and correlated problem. It is influenced and dynamically determined by a number of pre-conditions. The cost of a product design, for instance, will be strongly influenced by the quality of the product definition. Correctly understanding the customer's requirements and hence having a clear product definition will normally result in a quick and successful product design with a lower product design cost which is mainly determined by the total used engineers' hours (Tu et al. 2002). Likewise, the cost of manufacturing a part is normally affected by the design of the part. The final form of the part design, which includes the geometry, surface, tolerance and property specifications of the part, either directly or indirectly influences the selection of manufacturing processes and hence the cost of manufacturing the part. The bad experiences gained from manufacturing industries, particularly from sheet metal manufacturing companies, show that decisions made at the design stage without PD costing considerations normally result in a wrong product development. A wrong PD cycle will let the company suffer a longer PD cycle time due to a lot of reworks, a poorer product design and manufacture that is often indicated by a lower customer's satisfaction, and hence a higher PD cost.

Finally, I should point out that most developed or existing cost estimate and control methods and systems as reported in the literature have problems which cannot be directly adopted in the IRPD system. The detailed discussions on the shortcomings or limits of these developed methods and systems are given in Section 2, “Literature Review”, of this chapter. These methods and systems need to be modified or further developed so that they can be applied to address the complexity and uncertainty of a PD process in a global manufacturing environment.

The PD cost estimate and control is no doubt an effective tool to monitor a PD cycle. The cost estimate results can be used as important reference parameters or indicators to check the feasibility and profitability of developing a new product. It is well known from experience that reworks in down stream of a PD process, e.g. reworks in manufacturing, are more costly than correction made in earlier PD stages, e.g. in design. PD cost estimate and control is obviously a planning and control activity which has taken place at earlier PD stages. Hence to accurately estimate and optimally control the cost of a product development is crucial to shorten a PD cycle time and reduce the PD cost. To develop the useful methods, algorithm and system framework to support this planning and control activity at earlier PD stages is very important for OKP manufacturing companies to improve their market place, particularly for those small and middle sized OKP manufacturing companies.

This chapter, first starts with a review of the related research work and the discussions of limits and shortages of these developed methods, algorithms and systems as reported in the literature for PD cost estimate and control. Then, following this literature review, an Internet based system for PD cost estimate and control in a global manufacturing environment is proposed. This system was developed as a decision support tool for product developers to make decisions or sections among possible design alternatives, manufacturing processes and different ways of managing a product logistic chain under considerations of the PD cost and cycle time. To estimate and control the cost of a product manufacture and logistics, based on a product structure, a Cost Index Structure is proposed. To address the communication and data transfer problems in a global manufacturing environment, the STEP has been adopted in the Cost Index Structure to
model product data related to costing and relevant information through a PD life cycle. To demonstrate the principles of the presented system, data structure, methods and algorithms for PD cost estimate and control in a global manufacturing environment, a case study for developing a sheet metal part/product is discussed in this chapter.

6.2 Literature Review

To rapidly and economically develop a product has attracted a lot of attentions from industrial practitioners and academic scholars. Quite a few research papers have been found to address the problems of rapid and economic product development from the point of view of PD cost estimate and control, such as design for cost (DFC) and economical manufacturing (Ou-Yang et al. 1997), Neural Networks based costing and control (Bode 2000), cost estimation based on information management (Kals et al. 1999), concurrent costing (Ou-Yang et al. 1997), activity-based costing (Kolltai et al. 2000), etc. However, all these papers only addressed the problems in an individual manufacturing company which did not take the considerations of dynamic relationships among the manufacturing partners and the complexity of cost estimate and control in a global manufacturing environment.

As indicated by Becker et al. (1993), the accuracy or reliability of a PD cost estimate is increased along a PD progress. A PD cost estimate at earlier PD stages is normally very rough and not reliable. Sometimes, an over estimated PD cost of a product at the earlier PD stages may mislead a company into giving up a product which may have a great potential market. On the other hand, an under-estimated PD cost may trap the company in a nearly endless investment and end with a product that has a lower or even no market share. Hence to correctly estimate the PD cost of a product at the earlier PD stages is very important for rapid and economic product development.

To support decision-making at early PD stages, quite a few research efforts have been made for cost estimation at earlier PD stages. Typical approaches include parametric oriented costing, similarity-based costing, activity-based costing, detailed cost estimation, life cycle costing, learning curve costing and neural network based costing (Smunt 1999, Tamas et al. 2000, Bode et al. 1998 and 2000). Monden and Lee (1993) reported a cost estimate system called Kaizen Costing. This system was used for budgetary control and cost reduction of a product development. Horvath et al. (1994) proposed a standard costing system to meet cost limits (or standards) set by the management. Kawada and Johnson (1993) presented an accurate cost estimation system to support strategic management accounting. Cawthorne-Nugent et al. (1989) developed a computer aided costing system that employed the knowledge base technology and the cost models associated with part features. Marx et al. (1998) further reported another application of knowledge base technology for PD cost estimation. To solve the problem that at earlier PD stages there are not enough information and clear functional relationships between the product attributes and costs as well as the ambiguity of the prices or cost from suppliers, some artificial intelligence (AI) techniques and fuzzy logic technique, such as learning curves (Becker et al. 1993), neural networks (Tu et al. 1997a), and self-learn methodologies (Zhang et al. 1998) were applied in PD cost estimate.

Manufacturing cost is always an important factor to influence upon early decision-making on product design, process planning, product manufacture and final market share of the product. It is a major part of the overall PD cost. It is the cost of making the product. Usually, to determine the manufacturing cost of a product, a lot of detailed data will be needed, such data as selected manufacturing processes and the sequence of the processes, processing times of the processes, hourly rates for labour forces and machines, raw material cost, sub-contractor cost, transportation cost, logistic chain (i.e. supply and delivery chain) management cost, overhead cost, and equipment depreciation and maintenance cost. Traditionally, manufacturing cost is estimated through adding up the costs from these aspects or costs as aforementioned. The computer is often used as a computation tool to speed this adding up of calculations. Typical examples for applying this traditional manufacturing cost estimate method can be found in Casey (1987), Goldberg (1987) and Ostwald (1984). Winbourne and Toolsie (1991) proposed a manufacturing cost estimate system that can be used at the earlier product design stage to estimate the manufacturing cost of the product under
the uncertainty and ambiguity of the manufacturing process plan. A major drawback for this approach is the proposed system requires some initial estimates of the cost related product design and manufacturing parameters, e.g. surface treatment, machine set-up or tear-down times, etc. Like the aforementioned PD cost estimate research work, the knowledge base technology was also used in manufacturing cost estimate through storing the cost partners indexed by the cost-related product design and manufacturing parameters (Geiger and Dilts 1997). However, all these approaches did not tell how to correctly estimate the values of these cost-related product design and manufacturing parameters. Wang et al. (1995) presented a manufacturing cost estimating system, called the Totally Integrated Manufacturing Cost Estimating System (TIMCES), which integrated the computer aided design (CAD) system, process planning system and cost estimate system into one system. Inputs to this system include the selected manufacturing processes, the sequences of the processes, and cost data for standard components and manufacturing processes. It is obvious that the TIMCES is another application of the traditional manufacturing cost estimate method. It needs a large amount of complicated input data. Kamath et al. (1993) proposed a manufacturing cost estimate system for sheet metal parts/products manufacturing. This system in fact used the cost as an important parameter to select manufacturing processes and available machine tools with lower costs at every process planning stage. The system only considered the standard process sequence and 2-D shape of sheet metal parts/products.

In summary, all the aforementioned cost estimate methods or systems can be considered as applications of the following three basic cost estimate methods:

The function-based costing method:

This method estimates a PD cost through decomposing a product into generic or standard functional modules and quotes the costs for these generic or standard functional modules according to previous experiences or the market prices of these functional modules.

The similarity-based costing method:

This method predicts a PD cost through searching for cost information of similar products which were made in the past.

The expenditure-based costing method:

This method estimates a PD cost by listing all the possible expenditures associated with the PD cycle and quotes the costs for these expenditures based on the reference prices stored in a costing knowledge base. The key point in applying this method is to properly decompose a PD process into basic expenditures and these basic expenditures can be easily quoted according to standard market prices or previous experiences. The activity-based cost estimate methods or system are typical applications of this cost estimate method.

These basic cost estimate methods are also simply or in combination used in the PD cost estimate system for estimating cost of a manufacturing process, a component, or a consumed manufacturing resource. However, the major drawbacks to prevent us directly adopting the cost estimate systems as proposed in the aforementioned literature are: 1) All the research efforts as reported in the literature were focused on the problem in an individual manufacturing company rather than in a global manufacturing environment; 2) Nearly all the aforementioned research works assumed the processes through a PD cycle were carried out sequentially rather than concurrently as required by today’s small and middle sized manufacturing companies to rapidly and economically develop a product to meet a customer’s needs; 3) The proposed cost estimate systems are not able to dynamically and flexibly cope with some special problems in a global manufacturing environment, such as relatively frequent change of a product design and process plan due to the customer’s requirement change or the involvement of a partner or sub-contractor, a quicker price or cost change in a global market, higher uncertainty and ambiguity of product design and manufacturing processes.

-79-
at earlier PD stages for a product to be developed in a global manufacturing environment, as well as complicated cost estimate and optimal control through searching an available manufacturing resource in a global manufacturing environment.

In this chapter, a PD cost estimate system is proposed. It is a part of a rapid and economic product development system in a global manufacturing environment. This cost estimate system communicates with the other sub-systems or modules in the product development system through computer intra/internets. It was developed as a decision support system to help the selections of alternative design proposals, manufacturing processes, manufacturing resources and partners/sub-contractors from a cost optimisation point of view. As stated earlier in this chapter, to control a PD cost is in fact to speed up a PD cycle or shorten the PD cycle time. It interactively communicates and works with all the other sub-systems and models of the product development system to incrementally and concurrently develop a product feature by feature. To test the feasibility of the proposed cost estimate system, it has been implemented in a sheet metal manufacturing company to develop a sheet metal part. This industrial implementation is also reported in the later part of this chapter.

6.3 Overall System Structure of the Rapid and Economic Product Development

Figure 6.1 shows the overall structure of the rapid and economic product development system, which was developed for the small and middle sized manufacturing companies to quickly respond to and meet the customer’s needs. The system includes five sub-systems, i.e. Product Design, Manufacturing Process Planning and Optimisation System, Operations Management System, Cost Estimate and Optimal Control

![Diagram](image-url)

Figure 6.1 The overall structure of the rapid and economic product development system.
System, and Logistics Management System. These five sub-systems can communicate with each other via computer intra/internets. At earlier PD stages, these five sub-systems work together to generate an electronic file called Product Production Structure as shown in Figure 6.1. A product production structure is an integrated product data model, which can store all the necessary data through a product life cycle hierarchically in a common object. In this common object, the data are recorded in EXPRESS of STEP. Hence, all kinds of computer-aided engineering and management software packages should be able to operate this common object. This data model provides feasible solutions to build automatic links between the data of product design, process planning, logistics management (i.e. supply, partner and sub-contractor chains management), production scheduling and costing. For a detailed description of this product model, please refer to Chapter 7. In later PD stages, i.e. product prototype development and final production stages, these sub-systems as shown in Figure 6.1 will adaptively make changes to the Product Production Structure to cope with the possible changes from the customer, suppliers, partners, sub-contractors and the company itself. During the production stage, these sub-systems will work as optimal planning and control systems to control the production to stick to what were designed and planned in the product production structure.

In the product development system as shown in Figure 6.1, the Product Design sub-system mainly converts customer’s requirements into a product design, i.e. an engineering drawing plus product property and manufacturing requirement specifications. However, this design sub-system does not work like a traditional product design system which designs a product in one-go. It follows a concurrent engineering principle called “Prototype based incremental product development” (Tu et al. 2000c) to incrementally design a product feature by feature and concurrently search or plan all the necessary data for making the designed feature. This product design system consists of a computer aided customer interface which was developed through applying the QFD method to convert the customer’s requirements into the technical attributes of a product as addressed in Chapter 3. Based on the technical attributes, the search model will search for a similar product or similar products which were made in the past. The similar product(s) will be used as the prototype(s) of the product to be developed to meet customer’s needs. The product designer can design products through modifying the features of the prototype(s) or adding new features to the prototype(s) according to customer’s requirements in an incremental or feature-by-feature manner.

After a feature was modified or designed by Product Designers, the Manufacturing Process Planning and Optimisation System will search for suitable manufacturing processes from a GT (group technology) code index manufacturing process knowledge base or plan the processes through computer simulation technology or shop floor test if the processes cannot be found from the knowledge base (Tu et al. 2000b). It also consists of a manufacturing process optimisation model to optimise the manufacturing processes, e.g. tool selection and shortest path planning for a CNC cutting process. Through the Logistics Management System, it can search for alternative processes from the company’s partners and sub-contractors. As reported in Chapter 5, this process planning and optimisation system will plan alternative or optional manufacturing processes for creating the modified or designed feature of the product. Selection among these alternative or optional processes will be determined by the estimated costs of these processes, which will be provided by the Cost Estimate and Optimal Control System as shown in Figure 6.1. The finally selected manufacturing processes will be added into the Product Production Structure as part of the product data.

The Operations Management System will provide the information on production schedule, manufacturing resource availability or plan, and inventory of the raw materials or/and components, which is needed or relevant to a manufacturing process as planned by the Manufacturing Process Planning and Optimisation System. It should be pointed out that the Operations Management System manages the necessary manufacturing resources and materials not only inside a manufacturing company but also from the company’s suppliers, partners and sub-contractors via the Logistics Management System. The data managed and provided by this sub-system is in a global manufacturing environment. These global operations management data will be added into the Product Production Structure as a part of the product data together with the designed feature and planned processes.
Chapter 6 An Internet based Cost Estimation and Optimisation Framework for Rapid Sheet Metal Part/Product Development

The Logistics Management System consists of a data base called ‘Logistics Knowledge Base’ to record all the necessary data of the company’s suppliers, partners, and sub-contractors, such as their addresses, products (used as parts or components) or services (used as possible outsourced processes), prices, delivery times, quality/reliability/reputation (measured by ‘Excellent’, ‘Good’, ‘Average’, and ‘Poor’), manufacturability (e.g. special equipment, unique experience or process, special surface treatment, etc.), and transportation means and costs. This Logistics Knowledge Base can be accessed and retrieved via intra/internets (Tu 2000a). In addition to this knowledge base, the Logistics Management System also consists of an ‘Inquiry and Quotation’ process model, a web page management model and an automatic e-mail sending, receiving and processing system. The ‘Inquiry and Quotation’ process model can be used to formulate an inquiry on the company’s web page for purchasing a material, component, or a manufacturing process from possible suppliers, partners or sub-contractors globally. It directly links with the Logistics Knowledge Base. Through opening part of the Logistics Knowledge Base to the public, the possible suppliers, partners or sub-contractors can directly ‘write’ their responses into the Logistics Knowledge Base, which can be further sorted and processed by the Inquiry and Quotation process model to select the best supplier, partner or subcontractor. The web page management model is used to manage the company’s web page. The automatic e-mail sending, receiving and processing model manages and controls the company’s communications with its suppliers, partners, and sub-contractors. Normally, these e-mails are in a standard format, which looks like a data table but with hyperlinks. They will be automatically processed by the model and the necessary data will be extracted from these reply e-mails, such as company’s names and detailed addresses, costs/prices of the materials or processes, delivery times, transportation means, sample products or past works (in forms of electronic photos or engineering drawings) to show their ability to provide a wanted component or manufacturing process, etc. These logistics data will be added into the Product Production Structure together with the feature, manufacturing processes, and operations management data.

The Cost Estimate and Optimal Control System consists of a Cost Index Structure, a knowledge base and some cost optimisation models. It will be further discussed in the following sections.

The computer aided product development system as shown in Figure 6.1 can be operated by a product designer or developer to design a product, to plan how to make the product and to quote the cost of the product concurrently or incrementally in feature by feature manner. Since all the sub-systems as mentioned above communicate with each other through intra/internets, several engineers at different places in a company or the world collaboratively developing a customized product can also use this product development system. In my belief, this system is much more efficient and advanced than today’s commercial CAD/CAM systems which are widely used to support product design in the manufacturing industry.

Before discussing the Cost Estimate and Optimal Control System, I would like to define a PD cost in the following:

\[ \text{PD cost} = C_{dd} + C_p \]  (1)

Where:

\( C_{dd} \): Product Definition and Design cost.

\( C_p \): Product Production cost.

Product definition and design cost is determined by the consumed times of the product design engineers. To directly control these times does not make any sense since the main goal of product definition and design is to design a product with a high quality. A high quality product means the product will provide
maximum satisfaction for the customer. Hence a linear programming model was developed to maximize the customer's satisfaction under the constraints of customer's requirements, technical constraints, resource constraints and a given product definition and design budget (measured in dollars) which is normally decided by the company's management according to the company's marketing strategy and past experience (Tu et al. 2002). This means the product definition and design cost will not be estimated and controlled by the Cost Estimate and Optimal Control System. The costing sub-system in the product development system as shown in Figure 6.1 estimates and controls the total product manufactures and logistics cost.

The product production cost is the sum of the costs of all the manufacturing processes. The cost of a manufacturing process includes:

1) Test and possible rework fees due to the difficulty of the process.

2) The total consumption of the necessary manufacturing resources to carry out the process, including the consumptions of workforce, machines, and tools.

3) The total Logistics cost for purchasing and/or transporting a raw-material, component or outsourced manufacturing process, and relevant logistics management cost (e.g. communication cost and paper/data management cost). If a manufacturing process is outsourced or sub-contracted, it only includes logistics cost. If a raw-material or component is within the company, the logistics cost for this raw-material or component equals to the price of the material or component plus inventory and material handling cost in the company. In short, to simplify the cost estimate and control in a global environment, material handling and purchasing cost are considered as a part of the logistics cost.

6.4 Cost Index Structure

To estimate and control the manufacture cost and logistics cost of a product development, a Cost Index Structure was developed and illustrated in Figure 6.2. In accordance with the Product Production Structure which is a hierarchical common object data model, the Cost Index Structure adopts the same hierarchy as the Product Production Structure.

As shown in Figure 6.2, a manufacturing process will be associated with a cost index. A cost index includes three generic data classes, viz. Adding, Resource and Logistics, which are associated with the possible three costs of a manufacturing process as discussed at the end of Section 3 of this chapter. ‘Adding’ means adding cost to a manufacturing process. It is used to pay for the possible test and reworks due to the difficulty of the process. The difficulty of a process is determined by the design of a product, part, component or feature, e.g. geometry, surface finishing or treatment requirements, special tolerance and machining requirements, etc. Resource means resource cost or consumption of manufacturing resources as discussed in Section 3. Logistics means Logistics cost as discussed in Section 3. Likewise in the Product Production Structure, these cost data classes are written in EXPRESS of STEP so that they can be accessed and retrieved by different computer software systems.

The idea to build the Cost Index Structure in a hierarchy on one hand is to keep in line with the overall Product Production Structure. On the other hand, this is also a natural or logical way to estimate, control and record the cost of a product production from feature level on the bottom up to the product level on the top. It is obvious that the cost of a feature, which is added to a component, is a part of the cost of the component. Likewise, the costs of components, which are used to assemble to a part, will be a major part of the part cost, and the product cost will be determined by the costs of the parts that are finally assembled into the product. Hence, the Cost Estimate and Optimal control system estimates the cost of a product’s production from bottom level up to the top level of the hierarchy as shown in Figure 6.2. The models, e.g. Cost Estimate and Control models on the Component level, will try to estimate or search for the cost data (i.e. Adding, Resource, and Logistics) for the generic cost index associated with each manufacturing process.
by using the cost data from the cost modules on the lower level (e.g. Cost Estimate and Control Models on Feature level), the Costing Knowledge base, and the Logistics Management System. As shown in Figure 6.2, the Logistics Management System also includes a number of Logistics models which are hierarchically organized from the Feature level on the bottom and up to the Product level on the top respectively.

By the Cost Index Hierarchy as shown in Figure 6.2, a product production cost can be decomposed from level to level into detailed expenditures or indexes. In fact most of these expenditures on lower levels of the Cost Index Structure can be directly estimated by using the price of a standard component, part, or the cost of a common manufacturing process since to minimize the PD cost the design engineers always try to design the product by using as many as possible of these standard components, parts and common manufacturing processes. Also in practice, the costs of quite a few manufacturing processes normally on a higher level of the Cost Index Structure can be derived by the linear combinations of these standard prices or costs of common processes. Hence, a Costing Knowledge base was developed as an important part of the Cost Estimate and Optimal Control System to structurally record these standard prices and costs of common processes. However, it is difficult to propose a general data structure to build such a costing database for all kinds of manufacturing businesses. The data structure of a Costing Knowledge base for a particular manufacturing business (e.g. sheet metal manufacturing) is some how related to the special manufacturing requirements and practices in that manufacturing. Figure 6.3 shows a Costing Knowledge base that was developed for sheet metal manufacturing. It is obvious that this Costing Knowledge base does not fit for
mould/tool manufacturing.

Figure 6.3  A Costing Knowledge base for sheet metal manufacturing

However, the Cost Index Structure as proposed in Figure 6.2 is generic, which can be applied for all kinds of manufacturing. To be in line with this cost index hierarchy, the Costing Knowledge base needs to be developed to store the costing data in a similar hierarchical structure, i.e. the data tables in the data base are linked and indexed from a table which records the product data information on top down data tables which record the features data information on the bottom.

6.5 Cost Estimation and Optimisation

With the Cost Index Structure, the cost estimation and optimisation of the cost estimate and optimal control system can be described in the following two stages.

Stage 1: Cost estimate:

In this stage, the system needs to estimate the various costs of the cost indexes associated with the processes (See Figure 6.2) from the feature level on the bottom up to the product level on the top. The inputs to the system at this stage include manufacturing processes (including all the possible alternatives) planned by the Manufacturing Process Planning and Optimisation System (see Figure 6.1), manufacturing resources (including all the possible alternatives) and their cost rates and inventory records from the Operations Management System (see Figure 6.1), quotations or prices from all possible suppliers, partners and
sub-contractors which are searched and provided by the Logistics Management System (See Figure 6.1), and the costs or prices of these standard components and manufacturing processes from the Costing Knowledge base. With these input data, the Cost Estimate and Optimal Control system will estimate costs or instances for these generic cost indexes associated with the planned processes and their alternatives through using two cost estimate methods, viz. Generative Cost Estimate method and Variant and Knowledge based Cost Estimate method. These two methods will be further described in the following sub-sections. The cost estimate is an incremental or evolutionary process, which estimates the cost indexes feature by feature in line with the overall incremental product design, and production-planning process as implied by the concept of Prototype Based Incremental Product Development (Tu et al. 2000c). It is not developed to estimate various costs of making a product in one-go after a product has been completely designed and planned for production. This incrementally designs a product, plans the manufacturing processes, schedules manufacturing resources and inventory, manages logistics and estimates cost feature by feature from the bottom level of a Product Production Structure up to the top level and can provide the overall product development system with a strong flexibility to cope with the changes and uncertainties in earlier product design stages. Particularly, this incremental way of developing a product and estimating the cost can provide in-time feedbacks or references to support the engineers’ decision making on the selection of alternative designs, manufacturing processes, manufacturing resources and alternative suppliers, partners and sub-contractors. The shortages of this incremental method are the lack of global optimisation, and frequent and complicated communication management between these sub-systems as shown in Figure 6.1.

Stage 2: Cost Optimisation

Cost optimisation means a selection among alternative designs, manufacturing processes, manufacturing resources, inventories, suppliers, partners, and sub-contractors, or combinations of these alternatives with a lowest cost or total costs. Manufacturing process optimisation, e.g. to find the shortest path for a CNC cutting, and operations optimisation, e.g. an optimal production schedule with the shortest makespan, are not the tasks of cost optimisation. They are optimising tasks of other sub-systems as shown in Figure 6.1. e.g. the manufacturing process planning and optimisation system for sheet metal manufacturing includes a shortest CNC path planning model as addressed in Chapter 5, and the operations management for sheet metal manufacturing include an optimal production scheduling model (Tu et al.1997a). It is obvious that these optimisations will be carried out before the cost optimisation and they will help the cost reduction. For cost optimisation, an optimal algorithm was developed for the selection of alternative operation routines and supplier chains through using dynamic programming (Tu 1997c). More general, a commercial computer simulation package, e.g. ProModel (by ProModel Corp., Orem, UT), Quest and VNC (by Deneb Inc., Dayton, OH), is used to develop simulation models to solve cost optimisation problems. If a manufacturing process or a logistic chain consists of random variables or stochastic processes, Markov chains (Zijm 1984) may be used to model the process or the logistic chain and hence a cost optimisation heuristics or algorithm could be developed. This is now under research.

It should be mentioned that cost optimisation may be in conflict with other optimisations. An operational routine suggested by the Cost Estimate and Optimal Control system from the lowest cost point of view maybe in conflict with a production schedule suggested by the Operations Management System from the minimum makespan point of view. This in fact leads to a problem of decision making with multiple objectives, lowest cost vs. minimum makespan. The problem of decision making with multiple objectives can be solved by using rule based or heuristics/algorithms based methods, such as Analytic Hierarchy Process (AHP), Pareto Optimality and Trade off curves, goal programming, etc.

6.5.1 Generative Cost Estimate Method

Generative cost estimate method is a pump line method. With the Cost Index Structure as shown in Figure 6.2 and the input data from all the other systems as well as the price and cost data from the costing
knowledge base, the Cost Estimate and Optimal Control system can calculate the data or instances of the cost index associated with a process from the bottom level up to the top level of the Cost Index Structure. The cost index of punching a hole in sheet metal (a process on the feature level), for example, can be determined by calculating Adding cost, Resource Cost and Logistics cost. As mentioned before, the adding cost is decided according to the difficulty of the process, i.e. punching a hole in sheet metal for this example, and it is the sum of the costs of the possible tests and reworks. The resource cost for this example will be calculated by multiplying the standard cost rates of the used manufacturing resources (e.g. punch machine, tool and machine operator) to the used times of these manufacturing resources. The standard cost rates were recorded in the Costing Knowledge base and the used times can be provided by the Manufacturing Process Planning and Optimisation System. The Logistics cost is determined according to whether this process needs to take a material or component into consideration and how and from where this material or component can be provided. To avoid the duplicated cost data in these cost indexes, only the first process, which is among a group of sequential processes to be carried out on a blank or a sheet metal, will take the logistics cost of the blank or sheet metal into account. All the other succeeding processes will not take the logistics cost of this blank or sheet metal into account in their cost indexes under the Logistics, or a zero (0) will be assigned to the Logistics. However, if a succeeding process needs to add a new material or component, under the Logistics of its cost index, the logistics cost of this new material or component will be estimated and recorded. As mentioned before, the logistics cost is the sum of the costs of the purchased item, transportation (material handling plus inventory cost if the item is in the company), and communication. Normally, a supplier will quote the transportation cost as a part of the quotation for providing an item. Hence the Logistics Management System can provide the costs of the purchased item and relevant transportsations through inquiring for a quotation from suppliers. The communication cost for sourcing items depends on the difficulty of sourcing this item. According to previous experiences, the management of a company can set up some guidelines or controlled budgets for sourcing various items. These controlled budgets were recorded in the Costing Knowledge base and can be used by the Cost Estimate and Optimal Control System to quote or estimate the communication cost as a part of the logistics cost of getting a required item. If a process is an outsourced process, e.g. a sub-contracted process, it only has the logistics cost in its cost index and the other two terms will be zero.

When all the cost indexes have fixed data or instances, the production cost of a component, part or product can be calculated by the Cost Estimate and Optimal Control System through adding up all the cost data in the cost indexes associated with these processes which are under and linked (directly or indirectly) to the component, part or product in the Cost Index Structure.

6.5.2 Variant and Knowledge based Cost Estimate Method

As implied by the discussions made in the previous sub-section, the Generative Cost Estimate Method can only be used to estimate the costs of these standard components and processes. For those non-standard or 'new' components and processes that are required by the product design to address particular customer requirements, the variant and knowledge based cost estimate method needs to be employed. This method estimates the cost of a component or process through searching and comparing the similarities between the developing product and the products that were made in the past. As shown in Figure 6.1, all the products made in the history would have Product Production Structures that are integrated product data models. According to the discussions made this chapter, it is obvious that these product data models include cost indexes for all the manufacturing processes. In fact, after a product was made in the company, its final version of product data model (or Product Production Structure) was recorded in the product database which is the main part of the Design/Manufacturing Knowledge base of the product development system as shown in Figure 6.1. Through searching this Design/Manufacturing Knowledge base, the Cost Estimate and Optimal Control System may find the cost index of a process, which is in a past-made product production structure, is similar to the process of the developing product. The Cost Estimate and Optimal Control system search for a similar process by following a GT (group technology) coding system (Tu et al. 2000c) and the Product Production Structures are stored and retrieved in the
Design/Manufacturing Knowledge base according to the same GT coding system. The similar processes in the Design/Manufacturing Knowledge base have the same process codes. If a process of the new product is a combination of several primary processes, the Cost Estimate and Optimal Control system will search for the cost indexes of these primary processes and then determine the cost index for this combined process through adding up the cost indexes of these primary processes. A reference number rather than a GT code will be assigned to this combined process that is stored in the Design/Knowledge base as a part of the Product Production Structure. It will not be re-used or searched by the Cost Estimate and Optimal Control system or the Manufacturing Process Planning and Optimisation System.

6.5.3 Some Considerations on Cost Estimate

According to the experiences gained from this research and the industrial implementations, by using the two cost estimate methods as discussed in sub-sections 6.5.1 and 6.5.2, the Cost Estimate and Optimal Control System can estimate cost indexes for most planned manufacturing processes. However, there are still some processes that cannot be automatically estimated by using either method. Under this circumstance, the system will show these processes on the screen and ask the product developers to determine the cost indexes for these processes. For a combined process, if the system cannot find the cost indexes for some of its primary processes from the Design/Manufacturing Knowledge base, the system will show these primary processes on the screen, wait for a manual costing for these primary processes, and then decide the cost index for the combined process.

Another problem for cost estimating is that the cost of a component or outsourced process is within a fuzzy cost domain, i.e. from a lowest to a highest cost. Under this circumstance, a Neural Network-based algorithm plus some evaluation rules can be used to infer a most likely cost of the component or the outsourced process. Tu and Jiang (1997a) discussed this algorithm and the evaluation rules.

The third problem in cost estimation and optimisation is a conflict between cost optimisation and the reliability of suppliers. In practice, the cheapest cost normally implies an un-reliable supplier. The management of a company can solve this problem through choosing proper quality and marketing policies, e.g. only accept quotations from reliable suppliers, partners or sub-contractors.

6.6 Implementation of the Cost Estimate and Optimal Control System

The presented Cost Estimate and Optimal Control system has been implemented in a sheet metal manufacturing company. Through the development of a sheet metal part as shown in Figure 6.4, the main principles of the cost estimation and optimisation system are discussed as follows.

6.6.1 Case Scenario and System Description

As shown in Figure 6.4, the case part is a sheet metal part which is a part of a sheet metal product. To simplify the discussions here, I only show this sheet metal part instead of the whole sheet metal product. However, the readers can imagine that through a number of assembly processes this part will be finally assembled into the product.

This sheet metal part can be made by using CNC sheet metal punching machines or a CNC laser-cutting machine to punch or cut these holes, rectangles, and polygons on a flat sheet metal, and it is then formed by a bending machine into the final shape. The punching or cutting processes for making these holes, rectangles and polygons are indicated by F1 through F20 as shown in Figure 6.4. The part will be started as a flat sheet that only includes one face as indicated by P1 in Figure 6.4. After four bending processes, as indicated by B1 through B4 as shown in Figure 6.4, the final part will have 5 different faces, i.e. P2 through P5 as shown in Figure 6.4 in addition to P1.
Chapter 6 An Internet based Cost Estimation and Optimisation Framework for Rapid Sheet Metal Part/Product Development

The overall product development system as its structure illustrated in Figure 6.1 includes the following sub-systems:

1) Computer aided product definition system: This system was coded by using Visual C in association with Microsoft Excel. By using the QFD functions, the system converts the customer’s requirements into the technical attributes of the product as addressed in Chapter 3.

2) Computer aided product design system: ProEngineer is used as the computer aided product design system. However, a prototype searching model and an interface model were coded by C and integrated with the ProEngineer system through its API (application program interface). The prototype searching model is able to search for the prototypes, i.e. products made in the past, which have the most similar technical attributes to the product to be developed. These prototypes will be further modified into the customer wanted product according to the concept of Prototype Based Incremental Product Development. The interface model is used by the ProEngineer to read and write the Product Production Structure.

3) Manufacturing Process Planning and Optimisation System: This system was coded in C++. To meet the special requirements of sheet metal manufacturing, this system includes a nesting model and a CNC tool path-planning model in addition to the process-planning model. The nesting model can be used to automatically plan a cutting layout so that the trim loss can be minimized. The CNC tool path model was developed to optimally plan the CNC cutting or punching paths so that the total cutting time can be minimized. A screen printout of this system is shown in Figure 6.5.
Figure 6.5 shows a tool path planning process to be automatically carried out by the system. The parts as shown in Figure 6.5 have been nested by the system, i.e. properly placed on a given size of sheet so that the material waste or trim-loss is minimized. The part as shown in Figure 6.4 does not need the nesting optimisation but it needs the tool path planning so that these features could be punched or cut efficiently.

4) Operations Management System: It was coded in C++ and used to schedule shop floor operations and control the inventory through the communication with the company's MRP (manufacturing resource planning) or ERP (enterprise resource planning) system.

5) Logistics Management System: It was coded in Java. The HTML and XML languages as well as Common Gateway Interface (CGI) were used for writing the web pages and manage communication through computer intra/internets (Tu et al. 2000a).

6) Cost Estimate and Optimal Control system: It was code in C++ based on the cost estimate and optimisation methods and algorithms as discussed or mentioned in Section 5.

All these sub-systems can communicate to each other through computer intra/internets.
6.6.2 Cost Estimate and Optimisation for Making the Case Part

First, based on the generic Cost Index Structure as shown in Figure 6.2, the cost index structure for the case part can be drawn and illustrated in Figure 6.6. As shown in Figure 6.6, the part can be made by sequentially carrying out 24 punching or laser cutting and sheet metal forming processes. Some product states have been inserted into the manufacturing process chain. "P1" means the blank sheet which has one face P1. After the folding process B3, the semi-finished part will have two faces, P1 and P3. After the other forming processes, the part will have its final shape that includes 5 faces, P1 through P5. The Manufacturing Process Planning and Optimisation System determine the sequence of the processes as shown in Figure 6.6.

The first process, i.e. F1 as shown in Figure 6.6, will consist of the logistics cost in its cost index. The rest of the punching or cutting processes will not include a logistics cost if they can be carried out in one machine and do not need to be transported to other machines. The first folding process, B3, will include a logistics cost in its cost index. This is the cost for transporting the semi-finished part from the punching or laser cutting machine to a folding machine. The logistics cost of the rest of the folding processes will be zero if these folding processes can be completed in one folding machine.

In short, by using the methods (including the manual cost estimate) as discussed in Section 5, the cost indexes for all the manufacturing processes can be decided. After these cost indexes have been estimated, the Cost Estimate and Control System would carry out the cost optimisation if it were necessary. As mentioned in Section 5, the cost estimation is in fact a selection among alternative designs, manufacturing processes, or operation routines based on cost optimal (or cheapest cost) criterion.

First, let us assume that the machining features (i.e. F1 through F20 as shown in Figure 6.4) can be carried out in two alternative ways, i.e. 1) to use a CNC punch machine to punch all the 20 features, and 2) to use a laser cutting machine to cut the two polygons (F1 and F2) first and then use a punch machine to punch the rest of the holes and rectangles. The first alternative has the advantage of completing all processes
in one machine which saves the material handling cost between two machines, but it will need a special punch tool to punch these two polygons. The second alternative can let the company avoid having a special punch tool but it implies a material handling cost or logistics cost between the Laser cutting machine and the punch machine, particularly if this laser-cutting machine is an outsourced facility, or a sub-contracted process. Under this circumstance, if one ignores other technical factors and only use the cost factor as a reference to make a selection between these two alternatives, this can be done by using the Cost Estimate and Optimal Control system in the following way: 1) The Cost Index Structure as shown in Figure 6.6 is modified in Figure 6.7 which includes alternative processes; 2) The Cost Estimate and Optimal Control System estimates costs and decides the cost indexes for all processes; 3) Calculate the total costs for making the part by two process alternatives according to the calculation process as described in sub-section 5.1. Hence a decision can be made according to these two possible costs.

![Figure 6.7 Cost Index Structure which includes alternative processes.](image)

Second, if the part as shown in Figure 6.6 can be designed in another form, i.e. an alternative design, the Cost Index Structure as shown in Figure 6.6 will be modified to include alternative designs (or parts) on the Component level). Then these alternative designs will be decomposed into processes on the Feature level respectively. After this modification of the Cost Index Structure which can be simply imagined and is omitted from this chapter, the Cost Estimate and Optimal Control System will estimate the cost indexes for all the processes of the two different designs, and calculate the costs of the two alternative designs, just the same as it does to make a suggestion to select alternative processes as mentioned in the previous paragraph.
Third, if a part can be made by using alternative machines and through alternative operational routines as shown in Figure 6.8, the dynamic programming cost optimisation model in the Cost Estimate and Optimal Control System can be used to select an optimal routine to make the part which has the minimum total cost. Before the Cost Estimate and Optimal Control System can carry out the dynamic programming calculations, it needs to treat these processes on alternative machines as alternative processes, record them in the Cost Index Structure, and then estimate the cost indexes for all these alternative processes. Based on the cost indexes of these alternative processes, the system can calculate the cost between a pair of machines, which equals the machining cost on the next machine plus the material handling cost (or logistics cost) between these two machines. After all the costs between machines in the network as shown in Figure 6.8 are calculated by the system, the system will use the dynamical programming technique to find a shortest path from the Start point to the Finish point or an operation routine with minimum total cost.

![Diagram](image)

Figure 6.8 Dynamic programming network for making a part through alternative operation routines

This dynamical programming optimisation method can also be applied to select the cheapest logistics chain among a number of alternative chains if the machines as shown in Figure 6.8 are thought of suppliers, partners or sub-contractors. For a more complicated case in a manufacturing shop floor or a global manufacturing environment, the computer simulation technology, e.g. ProModel and Quest, can be employed for the cost estimation and optimisation. Tu et al. (2000b) reported an application of computer simulation technology to estimate and optimally control the cost of injection mould/tool making in a manufacturing shop floor.

6.7 Chapter Conclusions

The ability to correctly and rapidly estimate and optimally control a product development cost is always a strength which is pursued by manufacturing companies to improve their places in a keen competitive global market. Numerous research efforts have been made to automate the cost estimation and optimisation in a product development cycle although the research and development task is tedious and enormous sometimes. This chapter presented a feasible computer aided solution to automate the cost estimation and optimisation of a product development in a global manufacturing environment. By following the concept of the Prototype Based Incremental Product Development, it is able to provide in-time cost estimates and optimisation results to the product developers as important decision-making references in a
concurrent engineering approach of the product development. This concurrent approach enables product
developers to design the product, plan the processes, schedule operations on the shop floors, manage
logistics, and estimate and optimally control the cost feature by feature. This incremental way of developing
a product can help the product developers to clarify the product development life cycle data at earlier PD
stages and hence avoid reworks which often occur in the traditionally sequential approach of a product
development. To avoid reworks in a PD cycle, particularly for a customized product, will reduce the PD
cycle time and cost significantly.

The proposed Cost Index Structure is generic and can be applied to all kinds of product development.
It is modelled in EXPRESS of STEP as a part of the Product Production Structure. Hence it can be assessed
by all kinds of computer aided engineering or management software systems. It provides links or indexes to
associate various costs to product features through the manufacturing processes that are used to create these
features. In this way of recording the various costs of a product development, a cost analysis and
optimisation in a PD cycle can be easily and automatically done by the Cost Estimate and Optimal Control
System as presented in this chapter. This cost index structure as a part of the Product Production Structure
stored in the Design/Manufacturing Knowledge base can be further used as knowledge and references for
future product developments. It is very important for a manufacturing company to record its knowledge and
past experiences so that they can be reused to improve the company’s practices on product development.

The cost estimate methods and optimisation algorithms are effective. They can be easily understood
by and implemented in industry. The Cost Estimate and Optimal Control System as well as other
sub-systems of the product development system are running in an Intra/Internet communication
environment. This makes these systems or sub-systems run distributed in a computer network. It also
improves the system’s interdependence and reliability, and at same time reduces the complexity of these
systems’ design and development. To use the STEP and computer intra/internet communication technology
to develop these systems or sub-systems makes these systems very compatible with these commercial
computer aided engineering or management software systems, such as ProEngineer, SmartGroup’s PDM
(product data management) system, etc.
CHAPTER 7

AN INTEGRATED GLOBAL DATA STRUCTURE FOR SUPPORTING RAPID PRODUCT DEVELOPMENT

This chapter reports a research work that aims at developing an integrated data structure to support the RPD in the Internet environment. The emphasis is placed on the integrated data management and reuse of past product development experience to support a company's aim to shorten its product development cycle. The integrated global data structure model was modelled by using the EXPRESS of STEP with the consideration of real time data communication within the Internet environment. In terms of this data structure, a design/manufacturing knowledge base was developed as a major part of the WWW (world wide web) based product development system. The basic principles and concepts of the knowledge base and the WWW based knowledge management system will be presented in the chapter. An industrial implementation is also reported.

7.1 Introduction

Nowadays more and more manufacturing companies have realized that the ability to quickly develop a customized product through an economic and efficient way is critical for them to survive in the keen competitive international market. It has been widely conceived that the capability of RPD is one of the key issues that need to be considered to enhance the competitiveness of a company.

In order to quickly and successfully deliver new OKP products on to the market, the development of innovative products needs to be accelerated. This may require the support of new technologies (i.e. information technologies, Internet technology), in particular for companies whose business partners are distributed over the world. For the past decades, the computer and computer communication technologies have been widely applied to shorten the product development cycle, e.g. the use of powerful CAD technologies, information technology, optimization technology, virtual reality (VR) and virtual manufacturing technologies. Asynchronous Transfer Mode (ATM) networks and Fast Ethernet enable a quick and reliable exchange of relevant data and thus change the product development process tremendously. The Internet provides a new "platform" to access relevant information from all over the world in nearly real-time. Communication and cooperation between the members of a product development team who are located in different areas in the world, for instance, can be realized by computer supported cooperative work (CSCW) tools via the World Wide Web.

However, as addressed in Chapter 2, how these technologies can be integrated together to support RPD is still a problem. One of the reasons for not being able to solve this problem is that there is not a common integrated product data structure that can be combined with these new technologies and used in different stages of product development processes. Hence this chapter attacks this problem by presenting a global integrated data structure for supporting WWW based RPD. This integrated data structure has been used for building a WWW based knowledge base system to record and manage the data and information flow through a product development cycle in a New Zealand leading manufacturing company. The industrial test results show that the system can be used to support the product development process and effectively to shorten the product development cycle. This chapter mainly reports this research work in the following sections.
7.2 Overview of System Structure

The overall structure of the WWW based rapid product (industrial switches or control units) development system is shown in Figure 7.1. This system has been developed for a leading New Zealand manufacturing company to support and to speed up its product development cycle. In this system, the integrated data structure is used to build an environment for supporting the overall product development process, i.e., marketing, product design, tool making, and manufacturing. The integrated data structure can be used to build STEP-based product databases, and design and manufacturing knowledge bases. The WWW based information management system that contains these data and knowledge and data management software tools can be used to support the product development process, i.e., help product designers to make decision at the earlier stage of the product development process. Also, other application models (i.e., cost and simulation) can be achieved based on the integrated data structure. The methodologies that are used to achieve the system and the implementation of these application modules have been presented by Xie (2002a). The application program interfaces (APIs) of various computer aided engineering software systems (e.g., ProE) are the second development tools that can be used to develop the new application models (i.e., product fuzzy search model or supply chain management model). These models can be added into the open structured system. The information changes in different application models can be directly transferred to each other. As can be seen, the integrated data structure can be used to bridge the information gap between commercial software packages, user developed application software, databases and knowledge bases under the WWW environment. The structure of the integrated data environment will be described in the next section.

Figure 7.1 The Structure of the WWW based rapid mould product development system.

Figure 7.2 shows the home page on the RPD information management system. From the Figure 7.2, it can be seen that the system consists of seven computerized models, which can be accessed through the inter/intra-net.

The 'Project Appraisal' model is an on-line project management system. Each of the product developments will be treated as a project. The 'Project Appraisal' model records the managerial data about a project, such as the progress of the project (indicated by the achievements of milestones), product design engineer, tool design engineer, product manager, sales estimates, estimated project costs, and comments and endorsements of the departmental managers. These data will be real-time updated and can be accessed via the company's inter/intra-nets.
The 'Design/Manufacturing Knowledge Base' was developed according to the data structure to be presented in details in the following sections of the chapter. This knowledge base can be retrieved through inter/intra-net. It physically consists of several sub-data bases that are allocated in different departments, such as product design, tool design, tool making, shop floor, and inventory control. It records all the necessary information about the products which were manufactured in the past and the products under development, i.e. on-going projects. In fact, this knowledge base and its WWW based management system functions as a communication platform to facilitate the company's integrated data management and concurrent engineering approach at the earlier product design stages.

The cost optimization model was developed by using the dynamic programming technique and computer simulation technology (Tu et al. 2000a). This model estimates the cost for a product at its earlier design stages by optimally selecting possible manufacturing processes, sub-contractors, producing facilities, and shop floor scheduling through virtually manufacturing the product on a simulation platform, history data searching and seeking outside sourced services or sub-contracting. This cost model was reported in (Tu et al. 2000a, 2000b).

The 'Virtual Manufacturing' consists of the computer simulation models of the common CNC milling centers in the companies for product (mould/die) machining and a shop floor simulation platform. These simulation models were developed by using the VNC developed by Deneb Inc. (Dayton, OH) for CNC machine cutting simulation, and ProModel developed by ProModel Corp. (Orem, UT) for shop floor and production system simulation.
The ‘Customer Interface’ is an open interface for customers to get involved with the product development through the Internet platform. Through this interface, the customers can view the development progress of their products and give comments on their products. This is the only model that can be accessed by the customers. All the other models cannot be accessed by the customers.

The ‘Simulation’ includes two packages, i.e. VNC and ProModel. Through the ‘Simulation’, the staff in a company can access these two simulation packages to further develop some simulation models.

The ‘Computer Aided Process Planning’ model was developed based on a novel concept, called ‘prototype based incremental process planning’. This concept was presented by Tu et al. in (Tu et al. 2000a). This concept is a part of the prototype-based incremental product development method.

With the system as shown in Figure 7.1, the prototype-based incremental product development follows a certain pattern. First, the design targets (or primary features) that are identified by the user interface are prioritised according to their influence on customer satisfaction, manufacturability, quality, and cost. Correspondingly, different endeavours rates will be assigned to the design targets (or primary features). According to this prioritised list of primary features, from the design/manufacturing knowledge base, the system will automatically search for an existing product that is the closest to the required new product.

The existing products are the products that were successfully produced by the company in the past. The similarity between the existing product and the new product is measured by the sum of “similarity” rates of the common features between the existing product and the new product. The existing product with the highest sum of endeavour rates is taken as a prototype of the new product.

After a prototype is selected, a relevant process plan for producing this existing product is also found since they are stored in a extendable common object of the data model developed in terms of EXPRESS in STEP (See Section 4). Based on the prototype(s) and relevant process plan(s), an incremental process planning method as developed by (Tu et al. 2000a) can be applied so that the product design and process planning are carried out concurrently in a feature-by-feature stepwise manner. The incremental process planning method requires that if a new feature is added to the prototype, a sub-process plan for creating this new feature needs to be inserted into the process plan. If an existing feature of the prototype is modified, the process plan needs to be modified accordingly.

The sub-process plan for creating a feature can be searched from the existing products recorded in the design/manufacturing knowledge base or developed by the design/process-planning engineer through using his/her experience, a shop floor test, or a manufacturing process simulation package, e.g. Virtual NC. If a feature can not be created by any means, either the designer should avoid this feature or feedback should be given to the customer.

After a prototype or several prototypes are developed as described above, a new product will be designed, planned, and virtually manufactured on this system mainly by computer simulations and the data recorded in the design/manufacturing database. Finally, the output from this system is obtained as a fully recorded product data model.

7.3 STEP-Based Integrated Data Structure

As mentioned in the previous section, the product design/manufacturing knowledge base is created as an on-line data library to support all kinds of activities through a product development cycle in a desktop computer environment. It was developed according to a STEP-based integrated data structure. Figure 7.3 shows the framework of this data structure.
In Figure 7.3, the ‘Product’ means all the design and manufacturing data about a product. These data are physically stored in the computers in different departments and managed by the system according to an extendable common object data model. An extendable common object data model for an injection mould, for instance, is modelled by using the EXPRESS-G in Figure 7.4.

As shown in Figure 7.4, the product model is built as a tree structure. It can be envisaged that this type of tree-structure product data model can be well extended to meet the requirement of information modelling of different products (Tu et al. 2001). The integrated mould making data model as shown in Figure 4 already contains most commonly used components for a product manufacturing process. The product model in Figure 7.4 includes several entities on different levels. The top level is ‘Product’, which includes a number of sub-classes called ‘Components’ and a subclass called ‘Assembly’. Relevantly, all the necessary data regarding the components of the product will be recorded under the ‘Components’ subclasses. Likewise, all the necessary data about how to assemble the product will be recorded in the ‘Assembly’ subclass. A ‘Component’ sub-class includes

![Diagram showing the data structure of design/manufacturing knowledge base](image)

![Diagram showing the integrated product data model in EXPRESS-G for an injection mould.](image)
several ‘Primary_Features’ entities, several ‘Materials’ entities, and several ‘Secondary_Features’ entities. Accordingly, these entities record the primary features, materials, and secondary features of a component.

The primary features are those design and manufacturing features, by which the main shape, performance, function, manufacturability, and cost of a component is determined. For an injection mould, a cavity will be a primary feature. The secondary features are normally for the supporting functions and the integrity of the component (or product) or even the decorations of the component (or product), e.g. a screw hole, a slot, and a pocket for the parts assembly, the surface treatment of a mould, etc.

In Figure 7.4, either a ‘Primary_feature’ entity or a ‘Secondary_feature’ entity includes three types of sub-data-classes, viz. ‘Manufacturing_Constraints’, ‘Process_Plan’, and ‘Operational_Data’. The ‘Operational_Data’ further includes the ‘Schedule’ and ‘Cost’ sub-data-classes. Examples of using this data structure to model products can be found in (Tu et al. 2001) for an injection mould.

The ‘Suppliers’ as shown in Figure 7.3 includes all information about the company’s suppliers, i.e. contact information, product specification, reference prices, delivering lead times, reputation assessment, and quality assessment. The ‘Inventory’ records the company’s present inventory level, which is a part of the company’s MRP system. The ‘Tools’ records the company’s tooling information. The ‘Resources’ keeps the records of the company’s equipment and employees. The ‘Partners’ records the detailed information about the company’s sub-contractors. It contains important data on the evaluation of the sub-contractors’ manufacturability, reliability, and quality of their services or products. The ‘Operational_Data’ are the present and previous production planning and scheduling data records. This is a part of the company’s production planning and scheduling system.

7.4 The WWW based knowledge base system

Since efficient management of information flow in product development is critical to reduce the product development cycle time, new approaches and methodologies to achieve WWW based information management system have been continually put forward. CORBA (Common Object Request Broker Architecture), for example, is employed to ensure interoperability among distributed objects (Gruber et al. 1992). The STEP and KQML (Knowledge Query and Manipulation Language) are used to achieve the openness of the information system.

Figure 7.5 shows the structure of the WWW based information system that has been developed for a manufacturing company in New Zealand. This system comprises the WWW based software tools for information management and sharing among different departments, the integrated data environment and interfaces, the CORBA-based distributed environment, customer interfaces and several knowledge bases. The WWW-based database publishing tool and object communication tool were presented in (Xie 2002a).

To manage and share the distributed data and software systems/packages (or objects) in the company, the CORBA technology has been employed in the project to build a computer communication network or information sharing platform in the company (see Figure 7.5). In fact, in recent decades, the CORBA technology and the Distributed Component Object Model (DCOM) have been developed and applied for building such distributed object communication and management networks. Many successful cases can be found from the literature, e.g. (Mowbray 1995).

The customer interface was developed by using the QFD method (Tu et al. 2000b). It first provides an inter/intra network access for customers to input their requirements to the system. According to the customer’s requirements, the product designers can use the QFD models built in the system and the product design and manufacturing knowledge bases, which were developed according to the company’s previous experiences, to identify the technical features (or attributes) of the product to address the customer’s needs.
Chapter 7 An Integrated Data Structure for Supporting Rapid Product Development

A linear programming model was also developed to help the product development planner allocate the resources (mainly staff times) under the constraints of a fixed lead-time and a given product development budget to obtain various technical attributes of the product so that the maximum overall customer product satisfaction can be achieved. The system also provides the supporting functions for rapid product development, such as incremental process planning, manufacturing process simulation, shop floor operation simulation, cost estimate, and sub-contract planning and supplier selection.

![Diagram](image)

**Figure 7.5** the structure of the WWW based information management system

### 7.4.1. Knowledge base structure

In terms of the STEP based integrated data structure as discussed in section 4, a general product development knowledge base building framework has been developed for creating and managing the knowledge element, knowledge relationships and knowledge objects. The knowledge base building framework contains four information layers, which from the top to the bottom are knowledge layer, parts layer, feature layer and element layer. The ‘element layer’ contains the relevant data or information about the shape feature of a product. The ‘feature layer’ contains all the feature information that includes the feature attributes and relationships among the information objects in the feature layer. There information objects are defined by different production departments for a product development, such as feature geometric objects, manufacturing process objects, resource objects, and operation objects. The ‘part layer’ contains all the part information that includes the part feature information and relationships among different information objects in the part layer. The knowledge layer contains not only the part information, but also “knowledge related” information objects and a reference engine. Here, the knowledge is extracted from the part layer and the feature layer, and it is organized by information objects and relationships among
information objects. Therefore, the knowledge includes part-layer knowledge and feature-layer knowledge. A common information structure that contains the relationship of the information objects for a product development in a manufacturing company has been addressed in section 4. The knowledge in the knowledge layer can be directly used to support intelligent concurrent design and manufacturing. Here we add a management feature to the part features to manage all the information objects of a certain part. The management feature can be saved in a knowledge base and used as a part of knowledge data to manufacture the part. For different applications, the users can define their own application objects in part layer and feature layer through a graphic user interface (GUI) according to the requirements of a product development. After defining the application objects, the relationships among the objects can be created by the users or automatically by the optimization algorithm and existing knowledge. These objects with the defined relationships are the new knowledge, which will be added into the knowledge bases for future use.

7.4.2 Industrial Implementation

As mentioned in this chapter, the presented system has been implemented in a New Zealand leading electrical appliance manufacturing company. The company produces various industrial switches, control systems, and electrical appliances. In terms of the data structures and general system framework, the product design/manufacturing knowledge bases for rapid product development and the STEP based product databases have been built to support the company's rapid industrial switches development, which includes product development and mould/tool making.

In order to share the data and knowledge among the people in the company's branches which are located worldwide, the system has been built on a WWW based network environment. Hence, the WWW technologies and Internet software development tools, such as JAVA, CORBA, Visual InterDev, XML and KQML, have been used to develop this system. They will be further studied in Chapter 8. In the following, a case will be used to illustrate how the system works.

According to the customer’s requirements for a particular type of electrical switch, the searching
model in the system can find a prototype, i.e. a product produced by the company in history which has the maximum sum of endeavor rates. For this case, the prototype found by the system is 626VC industrial switch in the 600 series. The output from the system is shown in Figure 7.6.

In fact, Figure 7.6 shows a dynamic user interface on the web page. From this interface, a user can quickly search the necessary information for producing the 626VC. The user, for instance, can quickly get the CAD drawing of the product by double clicking "626VC" in the Drawing box (See Figure 7.6), the assembly illustration by clicking the "Product Assembly Picture" button on the right hand side of the page, designer's information by clicking the "Product Designer(s)" button, comments and discussions made at the earlier product design stages by clicking the "Product Design Questions" button, project appraisal information as mentioned in Section 1 by clicking the "Project Appraisal" button, and tool making information by clicking the "Tools that make product". The tool making information is shown in Figure 7.7.

![Figure 7.7 the tool interface](image)

Figure 7.7 shows the user interface for tool making information. From Figure 7.7, it is obvious that the product 626VC was made by using the injection mould M-5112. From this tool interface, a tool designer can get the relevant information for making a tool, such as cavities and parameters of the tool, other products made by using the same tool, factory molding machine data, manufacturing information (e.g. CNC machine, operator, NC code, etc.), tool components, tool materials (including the information about the suppliers), tool designers, and tool trials. For example to view general information on surface finish requirements on the tool M-5112 is simply performed by clicking the button directly located at the end of the surface finishing field box. The tool surface finishing requirements for mould M-5112 are shown in Figure 7.8.
In terms of this prototype and the customer’s requirements, the product designer(s) and tool designer(s) can quickly develop a new product and the necessary mould/tool through modifying the design and manufacturing information of switch 626VC and mould M-5112. Following the concept of incremental process planning as described in Section 3, the new design features for the product or the tool will be developed one by one concurrently with the pertinent manufacturing process plans. Some past experiences gained through developing 626VC were used as useful references to help the designers to avoid mistakes and reworks through the new product development cycle. This helps the company to dramatically reduce its product development cycle time.

The computerized models and the general knowledge base building frame as described in Section 4 were used to develop the knowledge bases or as planning or decision support tools through the product development cycle. All the relevant data and information through a product development cycle were recorded in terms of the extendable common data structure as presented in Figure 7.4. The information models behind the product interface as shown in Figure 7.6, the tool interface as shown in Figure 7.7 and the surface finishing information in Figure 7.8 are the instances or attributes of the common object 626VC which are extracted from several knowledge or product databases which are physically stored in different computers in the company. This WWW-based information management system automatically brings these data relevant to 626VC from these computers (or databases) via a front-end user interface (See Figure 7.6) to the users who have the right to access this communication network from anywhere in the world.

7.5 Chapter Conclusions

As indicated through the whole chapter, the main goal of this research work is to shorten the product development cycle. The research work as presented in this chapter made the following contributions to
achieve this research goal:

1) The WWW-based design knowledge base, which was developed according to the integrated data structure as presented in Section 4, can be used by a manufacturing company as an on-line data and information library. The engineers can easily and quickly source the necessary data and information for supporting a new product development. This data and information library can also help the company to well record and manage its past experiences.

2) The knowledge base management system as presented in this chapter can function as an inter/intra-net communication platform to facilitate and integrate the communications between engineers in different departments. The product design question in the product interface (see Figure 7.6), for example, is a record of such communications, particularly those made at the product’s earlier design stages. Normally in a manufacturing company, extensive communications involved at the earlier product design stages often result in misunderstandings and confusion between the departments due to poor management and record of these communications. These misunderstandings and confusion can easily prolong the product development cycle. By using the WWW-based knowledge base management system as presented in this chapter, these communications can be clearly recorded and easily accessed and viewed via the company’s inter/intra-nets. The engineer in a department can simultaneously view the comments made by the engineers in other departments. This will help the company to save times usually spend in those tedious cross-functional departments’ meetings.

3) As shown from the product interface (Figure 7.6) and tool interface (Figure 7.7), the knowledge base as presented in this chapter included a lot of descriptive information rather than the abstract data in a traditional database, such as product design questions, tool surface finishing requirements, designers’ information, manufacturing process and comments, etc. According to our experience with manufacturing companies, these descriptive comments are very important references to remind the designers and manufacturers to avoid making the same mistakes or spending time on a problem that was solved in the past. With these descriptive comments in the knowledge base, the knowledge base can be used as a training tool to help those graduate engineers to quickly grasp the company’s design and manufacturing experiences.

4) As described in Section 3 of the chapter, the integrated data structure was developed in terms of STEP. Since most computer aided design, engineering and management software vendors are supporting the STEP, the system as presented in this chapter should have a good compatibility with other computer aided software systems in advanced manufacturing.
CHAPTER 8

AN INTERNET BASED PRODUCT INFORMATION MANAGEMENT SYSTEM

As efficient management of product information that covers the whole lifecycle is critical to the enhancement of corporate competitiveness, this chapter explores the design and development of a WWW based product development information management system for a cross-nation manufacturing corporation that is headed by a holding company in Christchurch, New Zealand. Since product data are often managed in a distributed computing environment, CORBA is employed to ensure the interoperability among distributed information objects. This WWW based information management system will be discussed in this chapter, which includes two major components: 1) WWW based product design and development distributed object oriented databases; 2) A WWW based integrated system platform. Several sub-models are introduced, which include an object oriented database structure, a WWW-based information management system, a WWW database tool, an information access tool, the incremental process planning method and an integrated software platform for the integration of CAD, CAPP, CAM and the WWW based information management system.

8.1 Introduction

The widespread use of Internet and WWW has had a significant effect on the way of inter and intra communication of an OKP manufacturing company. Those efficient inter and intra net communication tools have helped OKP manufacturing companies to save the production cost, shorten the lead time to market, implement globalisation and concurrent engineering, and make them rethink all those issues at the heart of competitive manufacturing. In order to enhance the competitive ability, great changes are needed for OKP manufacturing companies, especially for those with “partners” globally distributed. It has been widely conceived that having the capability of rapid product development is one of the key issues needed to enhance corporate competitiveness. This may require the support of information systems, especially for companies whose business partners are distributed over the globe. A manufacturing corporation, which is managed by a holding company in Christchurch, New Zealand, is such an international manufacturing group. It has sub-branches in Australia and Malaysia. For this company, integration of various information systems through WWW is a holistic approach to manage the complexity in its product design, development, manufacture, and distribution, so that it can quickly respond to customers’ requirements.

At present, this company is facing an increasing demand by customers for product varieties, low cost and short delivery time. However, there are some hurdles in the company that impede meeting these customers’ requirements. First, at different stages of its tool/mould design and making process, suitable discussions are needed so that a tool/mould can actually be developed with respect to various manufacturing constraints. These discussions are normally made among different engineers in the company’s branches located in different countries through documented letters or email. They are time consuming and very inconvenient to manage. Second, quite a few important tool/mould making process data are not recorded and these data are lost after projects have been finished for a while. Besides the information lost in tool/mould making, delays, errors and long procedures of product definitions are also caused by lack of supporting information. Sometimes, the same product has to be redesigned owing to the loss of historical records.
Chapter 8 An Internet-based Product Information Management System

Hence, to develop a product information management system, which is able to record various data through a whole product development cycle and simultaneously provide an integrated platform for information sharing among different partners or departments, is very important for the company to shorten its product development cycle and cut down the production cost. A WWW based information management system for rapid and integrated product development has been developed. It consists of several distributed databases with object-oriented structure to store all the tool/mould making and product development information. Through the intra/inter-nets, all the information is published on the Web. This provides an easy way of accessing all the necessary information throughout the whole lifecycle of a product development process. However, the information accessibility does not mean that all the information can be used to directly support the design and manufacturing processes. Sometimes people will get confused when they face so much information and do not know what is useful for them. Therefore to integrate the available information is a key step for the different partners in a product development cycle to effectively share the information (Dong et al. 1998). In order to solve this problem, a great research effort has been made to develop an integrated information sharing platform in a computer network environment. Some prototypes have been developed, e.g. Dong et al. (1998), Xue et al. (1999), Cutkosky et al. (1993), Boynton (1993) and Gruber et al. (1996). However, for these prototypes, there existed some limitations to their applications, especially in a WWW environment. The first is the requirement for a formal WWW-based distributed database structure with suitable product data models for a manufacturing company, i.e. a decision must be made on what information should be shared and how to represent and record the information from a product development and the relevant tool/mould making processes. In practice, the company production manager usually makes his/her decisions about what information is needed, when it is needed and how it will be used depending on the context of the current problem (Boynton 1993). Also, the data structure of the product information management system may vary with the structure and culture of the company. Considering these questions in advance, collecting all the critical pieces of information for the decision, and finally integrating the information and decisions into a product model is usually a hard job (Gruber et al. 1996).

The critical issues in managing and sharing information for a cross-nation corporate are: (1) how to build the databases/knowledge bases with WWW publishing and real time information accessing mechanisms in a user-friendly operating environment; (2) how to develop a system, which can in real time capture related data and knowledge and use these data and knowledge to be shared or employed to support product development processes; (3) how to develop an integrated platform to integrate the information management system with existing software packages that are employed at different stage of the product development process. Boynton (1993) suggested that the critical issue for developing a design information system to support the company-wide design practice is not the product modelling aspect but the information dissemination aspect.

This chapter will discuss the problems as mentioned above through presenting a WWW-based integrated product development information management system. This system can directly support product design and manufacturing processes in a WWW environment.
8.2 The System Framework

The overall structure of the WWW-based information management system is shown in Figure 8.1.

![Image of system framework diagram]

Figure 8.1 Framework of the WWW-based integrated product development information management system.

The system was mainly build in two parts, viz. WWW-based information management system and the integrated platform. The first part includes distributed relational databases, STEP databases and knowledge bases, WWW database tools (WDTS), and user interfaces for different departments of the company to manage the product information. The second part includes a collaborative communication tool, an information access tool, incremental process planning (IPP) user interface and a cost optimisation model. This platform is achieved by using current WWW development tools (i.e. visual Interdev, JavaScript). The application tools are developed as agents that run in a distributed environment.

The distributed relational databases and knowledge bases record all the data and knowledge gathered from the previous product developments in the company’s history. The user interfaces were developed and can be viewed through Internet Explorer, which can be used by the different departments to change and manage the databases and knowledge bases on the intra or inter-net. The integrated platform was developed to integrate CAD, CAM, computer simulation packages, and the WWW-based information management system in order to support the so-called “smart drawing” and “smart manufacturing”. The incremental process planning user interface together with the cost optimisation model were developed to select or plan suitable manufacturing processes against design features from a cost optimisation point of view. The multi-objective optimisation algorithm of the cost has been discussed in Chapter 6. The information access tool was designed as a search tool to access all related databases and knowledge bases according to the requirements of the integrated platform. Some major models or tools as mentioned above will be further discussed in the chapter.
8.3 The Internet-based information Management System

Today's design and manufacturing practice in an OKP manufacturing company often involves complicated communication, interaction, and data exchange between individuals inside or outside of the company, such individuals as engineers, suppliers, and customers. For example, to design a part, a design engineer may need to access information on the process planning to determine the production requirements or the product data management system to find a similar part and revision levels. The challenge is how to create related information and knowledge so that they can be shared in real time. The development of intra/inter-net communication technology has provided a feasible solution to the company-wide knowledge sharing and real-time communication. Hence, a WWW-based distributed database and knowledge base management system is a fundamental part of the information integration of a manufacturing company.

8.3.1 Product Data Model

When data is added to a database, it becomes a model of that part of reality to which the data refers. As there is an increased need for up-to-date information, an automated data base management system (DBMS) was developed based on groups of formalized data modelling rules called product data models. Nowadays a product data model is usually object oriented. The EXPRESS data modelling language in the STEP has provided a useful tool to represent various product data, and the STEP has set up an international standard architecture for modelling a product. Hence the STEP and the object oriented modelling method were used in the research project to model the products as addressed in Chapter 7. Candaradi et al. (1994) also reported another application of the STEP to build a product data model. However, the STEP is still a developing data modelling architecture and it has some limitations for product data modelling, such as instance data or type data (Dong et al. 1998). It does not include a mechanism for using classification and inheritance for modelling products in a particular company (Manniso et al. 1998). Owing to the limitations of the STEP, the data model discussed in this chapter was built by both STEP modelling method and the traditional relational data modelling technique. For data that STEP cannot describe, related objects are created using object-oriented method and connect to its STEP object by using relationships. One of the advantages of using the object-oriented database (OODB) is that all the data in the object can be extracted by object identity or found by key. The newly created objects can be stored as well. The structure of the OODB and STEP database will be discussed in the next section. All data in an object can be easily read and written by using object oriented programming languages (OOPs) such as EXPRESS, C++ and JAVA.

8.3.2 Product Information Management System

8.3.2.1 OODBMS Structure

Figure 8.2 shows the framework of the distributed object oriented information management system.

![Diagram](image)

Figure 8.2 distributed information structure for a plastic mould design and manufacturing.
for a plastic mould design and manufacturing, which was developed for a manufacturing company in New Zealand. It represents a generic structure of an Object Oriented Database Management System (OODBMS) in the system. As shown in Figure 8.2, this information system includes product information, tool information, manufacturing information, and supplier information. These structures of information systems are built according to the requirements and the structures of the company, i.e. the product design database contains all the design specifications, geometric information and other information from the downside stream (i.e. manufacturability, cost). These information systems were built on an object oriented database structure and managed by using CORBA under the distributed computing environment. The detailed information with regard to how to use CORBA to integrate structured databases under multi-operation systems and a multi-language environment are well addressed by Fang et al. (2000), Kim et al. (2001a) and Mowbray (1995). Hence the distributed information management system eliminates the need to follow the predefined access paths to reach the target data, and makes the data access more flexible even under different operation systems. The structure of the information system facilitates uninterrupted queries and is well suited to the manufacturing environment. All the databases in the system were developed in ORACLE software package. ORACLE supports the large size of database with multi-user access, which is suitable for a distributed multi-user system. All the data in ORACLE databases can be accessed via ODBC and JDBC interfaces by using the Object Oriented Programming Languages (OOPLs) such as Visual C++, Borland Delphi and JAVA. This flexible programmable interfacing ability makes the databases more easily accessed and able to be further developed.

The object hierarchy of an Object Oriented Database that I developed for a mould/tool design and manufacturing company in Christchurch, which contains the detailed contents of the data class that created based on the structure of the information in different departments is shown in Figure 8.3. To support various types of data in an industry, different types of objects were developed. The structure of these objects and the relationships between the objects are very important for both the database management system and the system integration. For instance, it is usual that certain product design knowledge in a knowledge base needs to be associated with a product number or other identity number so that it can be correctly and effectively retrieved in the production. Figure 8.3 also shows several custom interface objects, such as product interface, tool interface and manufacture interface. These interface objects were developed for the

![Figure 8.3 the structure of an OODBMS for mould product development](image-url)
different departments in a company through using the Internet Explorer to access the databases according to the pre-defined privileges.

8.3.2.2 STEP Database Structure

Product data is one of the most important factors that need to be considered when deciding the dynamic changes of the product development processes. Under the WWW environment, as different CAD/CAM systems are employed by companies, the data sharing or exchange between different systems is very important to achieve rapid product development. Databases based on STEP can support product data exchange among heterogenous CAD/CAM systems. The proprietary file formats are not suitable for data exchange among different systems. STEP is promising in that it is becoming a new emerging standard for the exchange of product data throughout the whole lifecycle of products in distributed network environments. A STEP based information framework to cover the whole lifecycle of the sheet metal parts design and manufacturing processes has been presented in Chapter 4. This step-structure information modelling framework is well extended for other product development processes. A STEP knowledge base based on the information framework above is built for rapid mould/tool product design/manufacturing. Figure 8.4 shows a product information form that is based on the STEP based information framework structure above. Product designers and manufacturers can use this form to review and modify related information through Internet Explorer after it is published on the WWW.

8.3.3 WWW Database Tool

Although the database technology has been evolved for a long time. There is still missing an effective way to make the databases accessible through the intra/inter-nets so that the data in the databases can be shared on a global scale is still missing (Mannisto et al. 1998). Hence it is necessary to develop
WDTs that can publish and manage all relational data or knowledge bases over the Web. The WDTs developed by the author are used to automatically publish and manage the databases in a distributed WWW environment.

As shown in Figure 8.5, the WDTs were developed using several computer languages, which include those programming languages on the Web page, such as JavaScript, Hypertext Makeup language (HTML), Web authoring language, and CGI. CGI provides a standard protocol for communication between the client browser and the Web server. It was used to develop the dynamic data linkers between the Web server and the database management system for the different departments to access the databases through the intra/inter-nets. HTML language is used to design the Web pages. Visual C++ was used as a programming platform for developing the WDTs. Based on this platform, the programming entities, such as CGI linkers and HTML web pages, were coded and linked together. Through using C++, the WDT proxy, and the WDT context classes were also developed.

8.4. The Internet-based System Structure

Figure 8.6 shows the integrated platform, which is an intermediate layer between the WWW information management systems and the different software packages that are integrated into the platform as agents. This integrated platform is responsible for production optimisation and real-time information retrieving between the WWW based information management system and the agents.
This platform consists of several major components, i.e. a collaborative communication tool (CCT), an incremental process planning (IPP) user interface, a production cost optimisation model and an information access tool (IAT). By using this platform, all the software packages, IPP user interface, cost model and databases as shown in Figure 8.6 can be directly accessed through the intra-inter-nets.

In this platform, each agent manager is implemented as a CORBA object and exports certain public methods as an external interface. Other functions can also be achieved in the same way. For example, the detailed implementation method of a WWW data retrieve tool is shown as follows:

1. First define a CORBA object named Retrieve using CORBA IDL, its IDL description is as follows:

   Interface Retrieve {

   void retrievedata(in string dataFileName, in string schemaName, in string ProductName, out string ProductSpecification, out string dataFile, out string schemaFile, out string ProductContent)

   }

   Users can use function “retrievedata” to get the information needed through product name, dataFilename, etc. An interface is provided for users to input all the information for data searching. The code is written in C++.

2. Define the interface to the native code in the JAVA program, create corresponding .h file using JDK tool java.h. Then implement the native function using C++ and compile the C++ code into DLL. When the DLL is loaded, the JAVA program can invoke the function.

8.4.1 IAT and CCT

Many applications enabled by the WWW, such as virtual university, distance learning, electronic commerce, information gathering and filtering, have a strong need for tools supporting the effective retrieval of information. The goal of designing IAT is to provide a means for users to on-line search the data or knowledge according to a particular information enquiry in a product development process, and to real-time communicate with the information management systems via the intra-inter-nets. It was designed to be able to real-time control, retrieve and search for the data through the intra-inter-nets. The collaborative
communication tool (CCT) was designed for dynamically transferring messages, events and data in a WWW environment among the software packages, IPP user interface and cost model.

There still exist some problems in developing an IAT which can support concurrent multi-users applications, e.g. collaborative database accessing, distributed object technology, information retrieval, distributed services and resources management, etc. These problems will influence the communication speed between the OODBMS and the integrated platform. The search speed of IAT is very important for the system since it is obvious that a lower searching speed will influence the “dynamic characteristics” of the overall system integration. To accelerate the searching speed, the query-oriented search method and the user-oriented search method were combined to develop the searching model in this system.

As mentioned earlier in this section, the CCT was a tool to facilitate and manage the real-time communications in a WWW environment between the engineering software packages, the IPP interface and the cost optimisation model. It can dynamically capture data, messages or events and transfer them to the different engineering software packages, the IPP user interface and the cost model real-time through an intra/inter-net. For example, if a designer adds a feature to a product design by using a CAD package (e.g. ProEngineer), the CCT can through the intra/inter-net capture this change and transfer it to other software models, e.g. the IPP user interface, CAM package or computer simulation package, for further processing. The feedbacks can be also sent back to the design platform by the CCT. The CCT was developed by using the Pro/Engineer application programming toolkit (APT). The API of the ProEngineer toolkit consists of a library of functions, which were written in the C programming language. These functions can be easily incorporated in a program, which is written in C++. Since the CCT was developed on the ProEngineer APT platform, it can run on the ProEngineer operating platform.

8.4.2 Incremental Process Planning (IPP)

Tu et al. (2000) presented an incremental product design and process planning strategy for developing customized products or OKP products. This strategy suggests the company to plan the manufacturing process to each of the new or modified features, which are added to the design of a product. The features, which can not be feasibly or economically manufactured, should be changed or given up. They also suggested that because of a great variety of OKP products and a usually high demand from the customer for a lower cost and a shorter delivery time, the history data and knowledge reuse is very important to a OKP company to successfully develop an OKP product.

For the WWW-based product development information management system, the IPP method also plays an important role since the IPP strategy can help a manufacturing company to effectively avoid production reworks and the history data and knowledge reuse has been approved as an effective way to cut

![Diagram](image)

Figure 8.7 information flows among the different models
the product development cost and lead-time (Tu et al. 2000a). The IPP user interface was developed for users to plan the manufacturing processes according to the IPP manufacturing strategy. It interacts and communicates with the data or knowledge bases and other models via intra/inter-nets. The information flows among these models are illustrated in Figure 8.7. As shown in Figure 8.7, the IPP user interface can get information from the WWW database management system, and send the possible manufacturing process plans to the cost optimisation model. After getting the optimisation result from the cost optimisation model, it can save the result into the databases. The IPP user interface can also dynamically communicate with CAD and CAM packages through the CCT.

8.4.3 Cost Optimisation Model

A generic cost estimation and optimisation framework has been reported in Chapter 6. Different from that framework, the cost optimisation model was developed to save the product development cost through optimally or rationally scheduling the production operations in shop floors. The inputs to this model are possible manufacturing process plans, which were planned by the process planning engineers by using the IPP user interface, and the capacity availability of production resources, which are normally recorded in a company's MRP system. The outputs from the cost optimisation model are a production schedule and a relevant cost estimate. The cost optimisation model was developed by using case based reasoning method, probabilistic dynamic programming and computer simulation technology. For different products, the cost optimisation function may be different. However, the optimisation algorithm will have something in common, i.e. the cost optimisation model of sheet metal products cutting has been transferred to a common mathematical model of NP completeness problem (Xie et al. 2000). A mathematical model has been built for the mould products cost estimation and optimisation. The basic cost of a product can be estimated and optimised while considering several possible manufacturing plans. The model can also quickly send the design engineer a cost estimate by using case reasoning method to search for a similar solution from the databases via an intra/inter-net. It can also derive a cost estimate and a production schedule to a designed feature by using probabilistic dynamic programming technique and computer simulation technology to virtually schedule and manufacture the part or the product (Tu et al. 2000b).

8.5 System Implementation

Key challenges still exist in the development of WWW-based support systems, such as intelligent search engine, Web accessibility, collaborative and distributed application environment, scalability of Web servers, intelligent agents, server security, the limitations of the programming languages like HTML, XML, etc. All these challenges have to be considered while developing the WWW-based information management system for rapid and integrated product development. They will influence the functionality, reliability, and the realization of the integrated system. In this section, I aim to demonstrate the main functions and the feasibility of the system by briefly describing a prototype system, which was developed for a New Zealand manufacturing company to support their mould products development.
Figure 8.8 shows the overall structure of the prototype system, which includes several components.

Figure 8.8 the WWW-based integrated information management system for mould products design and manufacturing.

This system can be extended and new functions can be easily added and integrated into the system through the agent technology. A CNC machine simulation package called Virtual NC was used to develop the manufacturability test bed and a shop floor simulation package called ProModel was used to develop the virtual manufacturing shop floors for the cost optimisation model. These software packages can be used as real time simulation for product manufacturing process. The simulation result will be recorded in the WWW based information system. This information can be retrieved for decision support for similar product development. A Web client called Pro/INTRALINK in the ProEngineer makes it possible to access or manipulate the product data through a Web. Specific interfaces have been designed for people in different departments to activate functions that are provided by the integrated platform, i.e. change/search information or see the manufacturing simulation process. C++ is employed to code all the software models as described in the chapter, such as the IPP user interface, the WDT, the CCT, etc. These tools can be used to locate related information and can achieve intelligent searching functions. With the integrated platform that contains these tools, the WWW based information management system can be used in different stages of the product development processes, i.e. at the earlier design stage, the designer can search existing WWW based database/ knowledge bases for existing similar products through a WWW based interface that is designed for product designers, this knowledge or information (i.e. cost information, lead time, manufacturing ability) can help the designer to make decisions. Also, the integrated platform supported by the WWW based information management system can be used as a concurrent product development platform through information integration and sharing at each stage of the product development processes. Instead of following the traditional, sequentially arranged product development process, the system can incorporate considerations such as manufacturability, assemblability, serviceability and recyclability into product design or planning stage. The time consuming reworks can be avoided and thus the product development time or
cost can be cut.

8.6 Conclusion of the Chapter

It has been well recognised that the development of WWW technology provide an efficient tool and a very promising direction for manufacturing companies to revolutionarily change their way of managing and integrating information flows in an enterprise. This chapter presented a framework to develop a WWW-based information management system for rapid and integrated product development. A prototype based on this framework has been developed and implemented in a New Zealand manufacturing company to support their mould product development. The structure of the proposed system reflects two aspects: the basic structure of information sharing and the integrated environment for task execution (i.e. optimisation). Our initial experiments during the development of this system with database management system, automatic email transferring system, manufacturing workflow systems and the cost optimisation model have been very encouraging. The manufacturing company involved in this research project has adopted the system in the mould/tool products development process. Problems existing in the company as mentioned in section 1 have been solved. The product development cost has been greatly cut and product development lead-time has been shortened.
CHAPTER 9

INTERNET BASED DFX FOR RAPID TOOL/MOULD MAKING

Computer Internet communication technology offers tremendous potential for building computer communication and software platforms for rapid development of OKP products to meet global competition. In the past few years, a variety of Internet based systems have been developed for the purpose of RPD. Among these systems, Internet based "Design for X (DFX)" systems have been recognized as efficient tools for implementing concurrent engineering and playing a key role in RPD. Internet based DFX or IDFX systems can be applied by manufacturing companies to rapidly produce high quality products with low costs and thus higher profits. However, the implementation of IDFX systems is not an easy task. The reason is that many new techniques are involved in the application of the IDFX and the PD processes are normally sophisticated and vary with product development environments. In this chapter, as illustrated by case studies, two typical applications of IDFX, i.e. Internet based Design for Manufacture (IDFM) and Internet based Design for Cost (IDFC) systems, are proposed for rapid and economical tool/ mould making. The structure and the key models of the systems are discussed.

9.1 Introduction

DFX has been recognized as one of the most effective approaches to implement concurrent engineering for the goals of RPD (Xie et al. 2002a). Efficient DFX tools can be used to gather and present facts about products and processes, to clarify and analyse relationships between products and processes, to measure performance, to provide redesign advice on how a product design and manufacture can be improved, etc. The advantages of implementing DFX can be summarized as: 1) It provides measurable competitive improvements, which include improved quality, compressed cycle time, reduced life-cycle cost, increased flexibility, improved productivity, and enhanced customers' satisfaction; 2) It improves and rationalizes decisions on product design, process planning and resource deployment; 3) It provides a far-reaching effect on operational efficiency in product development (Huang 1996).

The Internet technology is also becoming popular for the DFX analysis (Wagner et al. 1997) as it can bring tremendous benefits for manufacturing companies, especially those with partners that are globally distributed. Internet based DFX services can be used to support concurrent, collaborative and economical product development, thus reducing the time and cost of product design, assembly and manufacture. Using Internet to deliver DFX functions can help to achieve better communications, closer cooperation, concurrence, transparency, improved customer and supplier involvement, easier project management, team-building in design work, and rationalizing and structuring product development process. For example, Internet based DFA technology can be used to reduce the cost and time of assembly in a distributed manner by simplifying the product and process by means of reducing the number of parts, combining two or more parts into one, reducing or eliminating adjustments, simplifying assembly operations, designing for parts handling and presentation, selecting fasteners for ease of assembly, minimizing parts tangling, and ensuring that products are easily tested. IDFC (Internet based design for cost) systems can help to eliminate those expensive and unnecessary features of a part which leads to an increased product manufacture cost. IDFM (Internet based design for manufacture) systems can help to determine the manufacturability of the possible designs, manufacturing process, machine tools, etc. at early design stages. The application of IDFM systems can help to eliminate the unnecessary features of a part, which are often difficult or impossible to manufacture. The impact of Internet based DFX or IDFX systems can be found through the whole product
development life cycle, e.g. applying Internet based DFA to reduce the number of parts and hence reduce inventory and inventory management effort.

Although there are so many advantages of developing IDFX systems for RPD, the implementation of IDFX systems is never an easy task. Several problems have been raised in recent years for implementing DFX systems. Among these problems, an important question is whether there is a basic pattern for the development of these DFX tools (Olesen 1992). Another important issue is how to update traditional DFX systems to meet the requirements of the global RPD environment. These problems have become the major hurdles for developing and implementing the IDFX systems.

9.2 System Structure

Figure 9.1 shows the structure of the Internet-based DFX system. The Internet based DFM system consists of two major components. The first one is the Internet-based DFM data/knowledge base that contains mould product design for “X” information and knowledge. The DFX data/knowledge base is modelled according to a modelling methodology that will be addressed in section 9.3. The DFX data/knowledge base is organized in appropriate forms for easy access. The other one is the Internet based prototype system to deliver the DFX functions through the Internet. This prototype system was developed for a mould manufacturing company, which contains three sub-systems, viz. client system, server system and DFX knowledge system.

The DFX knowledge system is a Microsoft Windows NT server computer that runs a product data management system (Tu et al., 2000a). The product data management system is stored in a SQL server, which can be accessed from other servers or clients by using designed interfaces and tools. The main function of the product data management system is built based on the DFX principle, which contains all the guidelines of the DFX (mainly DFM and DFC) for plastic injection moulds development.

The server system is also a Microsoft Windows NT server computer that runs the Web server software systems such as Internet Information Server (IIS), Blazix Java Web-Server and EJB-Server (http://www.blazix.com/). These servers are used to provide the Internet services. There is a virtual directory in the Web server. All the Web applications of the software systems (e.g. ASP file, HTML file) are stored into the virtual directory and these files can be accessed from either inside or outside the company. In this system, I have developed Internet publishing tools by using Visual Basic for automatically publishing DFM guidelines on the Web as addressed in Chapter 8.

The client system is a local computer that runs Internet Explorer or other Web browser software tools. The clients are usually members of the product design and production team. They can use the client system to real time exchange ideas and discuss the issues in the product development process. Also they can
formed new DFM data knowledge according to given priorities and they can be used as new guidelines for designing and manufacturing similar products.

The IDFX systems also contain several customer interface objects, such as product interface, tool interface, and cost interface and manufacture interface. These interface objects were developed for the different functional departments in a company through using the Internet Explorer to access the DFX knowledge databases according to the pre-defined privileges. Some software tools are developed to support the Internet based DFM system, e.g. communication tools among these three sub-systems, data modelling and searching model, and customer interface. These software tools were well addressed by Tu et al. (2000).

9.3 DFX Knowledge Base Structure

9.3.1 Knowledge Base Structure

As mentioned in section 9.2, DFX data/knowledge base is one of the important components of the IDFX systems. The DFX data/knowledge base has to be built in an appropriate form so that the DFX information and knowledge can be easily managed and reused as guidelines. This section addresses how to build up the DFX knowledge base through creating a distributed DFM knowledge base in an injection mould manufacturing company. Figure 9.2 shows the structure of the DFM data/knowledge base. It represents a generic structure of an Object Oriented Database Management System. As shown in Figure 9.2, this DFM data/knowledge base system contains the product information, tool information, manufacturing information, and supplier information. The structure of DFX knowledge bases built according to the requirements and the structures of the company, e.g. the product design database contains all the design specifications, geometric information and other information from downside stream (i.e. manufacturability and cost). These DFX knowledge bases were built on object oriented database structure and managed by using CORBA (Fang et al. 2000) under the distributed computing environment. The detail information with regard to how to use CORBA to integrate structured databases under multi-operation systems and multi-language environment are well addressed by Fang et al. (2000) and Kim et al. (2001b). Hence the DFM data/knowledge base can be used in a distributed environment. It eliminates the need to follow the predefined access paths to reach the target data, and makes the data access more flexible even under different operation systems. This data structure facilitates uninterrupted queries and it is well suited to the global manufacturing environment.
In the similar way, other DFX knowledge bases (e.g. DFC knowledge base) are developed for rapid mould development based on an object-oriented and distributed manner. Usually, DFX knowledge bases consist of the detailed contents of the data class that was created in terms of the structure of the information in different departments. To support various types of data in industry, different types of objects were developed based on a generic DFX model that will be addressed in the following section. This generic DFX model is very important not only for structuring DFX systems but also for integrating the system with other product development systems, e.g. CAD system, product data management system and enterprise resource planning system. For instance, it is usual that the product design knowledge in a knowledge base needs to be associated with a product number or other identity number so that it can be correctly and effectively retrieved in the production.

9.3.2 Generic DFX Model

The need for a basic DFX pattern is essential for the development of IDFX tools. For example, a generic DFX model would speed up the development of specific DFX tools dramatically. It can provide a platform for integrating multiple DFX tools to facilitate the flow of data and decisions between these DFX tools or systems. It also provides a platform for integrating a DFX tool with other decision support systems used in product development such as CAD/CAPP/CAM and CAPM (Computer Aided Production Management), and facilitates the flow of data and decisions between the DFX tool and the other systems. Furthermore, a generic DFX model can provide a common basis on which a trade-off can be assessed between competing issues when multiple DFX tools are used. The advantages of establishing a generic DFX model can be further viewed from other prospects, such as speeding up the development of DFX systems and helping to select appropriate DFX tool for a problem (Huang 1996). In this chapter, a DFX data model based on the information framework proposed in Chapter 4 is discussed.

In order to make the DFX (DFM/DFC) systems more compatible with other product development software systems, a generic DFX data modelling methodology is needed for modelling DFX (DFM/DFC) data and knowledge. In the Internet environment, product data is one of the most important factors that need to be considered in the case of dynamic changes of the product development processes. As there are different CAD/CAM systems employed in most manufacturing companies, the data sharing or exchange between different systems is essential to achieve RPD. DFX data model based on STEP can support product data exchange among heterogeneous CAD/CAM systems. The proprietary file formats do not meet the need for dynamic data exchange between different systems in a global environment. The combination of STEP and Internet based data model solutions (e.g. CORBA) provide a solution to solve the product data exchange problems and it is becoming a new emerging standard for the exchange of product data throughout the whole lifecycle of the product in a distributed network environment. Tu et al. (2001) presented a STEP based information-modelling framework to cover the whole lifecycle of the mould design and manufacturing processes. This information-modelling framework is called step-structure information modelling framework, which contains four top-down information layers, viz. knowledge layer, parts layer, feature layer and parametric layer (Tu et al. 2001). This framework was first used for information modelling of sheet metal parts and has been extended for modelling mould and mould development processes. The DFX data model of a mould can be built by following this information framework, which includes the following layers: DFX knowledge layer, DFX product layer, DFX feature layer and DFX parametric layer. The DFX parametric layer contains the geometric data of the shape feature of a product (or mould) in terms of “X” information, e.g. the manufacturing information for a special feature. The DFX feature layer contains all the feature related DFX information data that include not only the feature information (or attributes) but also relationships between the “X” information objects on the feature-level. The DFX part layer contains all the part information data that include down-level DFX feature information and the relationships among part-level DFX information objects. The DFX knowledge layer contains not only the DFX part information, but also “DFX knowledge related information, as well as an inference engine. The knowledge in the knowledge layer is extracted from DFX part-level knowledge and DFX feature-level knowledge, which are formed by information objects and relationships between the objects.
9.4 DFX via the Internet

This section will discuss how to implement two important DFX systems for rapid and economical development of a plastics injection mould, i.e. the IDFM system and the IDFC system. The method and process of developing these two IDFX systems are in fact generally applied for the development of other IDFX systems, such as an IDFA system, an IDFR (internet based design for resource) system.

9.4.1 IDFM for Rapid Mould Development

In order to support rapid mould development, an IDFM system for a mould-manufacturing company in New Zealand has been developed. Figure 9.3 shows some screen-prints out from this system. This system can achieve the DFM functions through Internet access. It can be used to develop modular products that are normally assembled by common components, e.g. a plastics injection mould. It helps to prevent these unnecessarily high feature design and manufacturing requirements, such as high smooth surface requirements, the radius that is unnecessarily small, and the tolerances that are unnecessarily high. The objective of reducing the number of parts in the DFM may lead to highly integrated, complicated, and multi-functional parts. On the other hand, the DFM also aims to keep an individual part simple, because extremely complicated parts can hide a high cost that is not initially apparent.

As shown in Figure 9.3, the IDFM system mainly includes the following functions:

1. An Internet based DFM data/knowledge base that contains specific guidelines organized in an appropriate form for easy access, which is addressed in section 9.3.

2. Internet based tools for data operations (e.g. edit, search and retrieve) of the DFM data/knowledge base and other databases.

3. Various DFM checklists and guidelines, which provide examples of good practice, and remind the designer to check various manufacturing requirements and constraints. Product designers can easily get the knowledge of a product design and manufacture by checking the manufacturing information from the product ID. The Internet based DFM system is also supported by a computer software system that is able to score a product in terms of the ease of its assembly, assembly cost and

---

Figure 9.3 DFM via the Internet
time.

The DFM guidelines can be automatically reorganized and reformed according to the sub-components of a new mould (or product). This will allow the manufacturing company to meet the dynamic changes in its global market, and at the same time reduce the workload of the product manufacture.

Based on the generic DFX model discussed in section 9.3.2, a product data model is built based on the "common object concept". This data model can be used to model various moulds and can be accessed by using various product development tools (e.g. a CAD/CAM system a Product Data Management System, an Internet based logistics management system, a computer aided management system, etc.). This common object is a part of the STEP based data model that is used to model various products. The information that is included in the data model is also stored in the DFM knowledge bases. As described by Tu et al. (2001), a product that is modelled in this system is called 'Soft product'. The soft product in the Internet based DFX systems provides a detailed roadmap for manufacturing a physical product, e.g. an industrial switch as shown in Figure 9.3. The physical product is called a 'Hard product' by Tu et al. (2001). It is obvious that if a soft product is developed, a hard product can be rapidly produced in the company.

Through implementing this IDFM system, the company has reported the decrease of the costly redesigns, design errors, manufacturing costs and cycle time. The structure of the IDFM system can be easily extended for developing other DFX systems, e.g. an IDFA system and an IDFC system. The DFA technology is closely linked to the DFM technology, but is oriented primarily to assemblies, semi-assemblies, and the final product.

From the discussions above, it is obvious that product developers should be able, through IDFM systems (including related product data management system), to access information that is useful for them to improve the design of the part. This is very important for rapidly developing new products. The IDFM system can be called up to analyse the current state of their designs, point out where the design is too complicated, and indicate possible areas of improvement. This is critical at a product design stage, especially at the earlier design stage. The DFM particularly is best performed at conceptual product design stage, i.e. before major decisions about product and process characteristics have been finalized. At this early stage of the design, there may not be much information to work with, but the IDFM systems will make sure that all the existing information can be made available to a product design team, of which may be globally distributed.

9.4.2 IDFC for Economical Mould Development

This section presents an IDFC system developed to help the mould designers predict the costs of manufacturing injection moulds and enable them to make more cost effective design decisions. The main goal of the IDFC system is to estimate the product development cost during the early design stages through achieving the following objectives:

1) To find out the segment of a part model that may cause a high product development cost;

2) To provide an environment to estimate alternative costs of possible designs;

3) To automatically calculate and optimise the product development cost and help decision making at the early product stages.

In order to develop an IDFC software tool for early cost estimation, the steps required to estimate the manufacturing cost need to be identified. Figure 9.4 shows the cost estimation steps of developing an injection mould. Different from what is discussed in chapter 6, the IDFC discussed in this chapter is more specific in cost estimation and optimisation of developing a mould that normally consists of basic common
components. Obviously, its cost optimisation and estimation methods are different from the ones used in sheet metal part development as described in Chapter 6.

In the IDFC system, the costing processes normally consist of the following steps. First, to estimate the mould development cost, the mould making processes need to be decided. The mould making processes are decided according to the mould design specifications. Then the cost of making a mould is estimated by

![Diagram](image)

**Fig.9.4 Basic structure of IDFC for developing plastic injection moulds**

adding up the costs incurred in these mould making process as well as the material costs. Usually, the cost estimate of making a plastic injection mould requires the incorporation of raw material, labour, machine set-up and operating costs, and overheads. In real life, mould cost is estimated according to rough plastic part features or merely a mock-up (i.e. without the details of mould design) at the early design stages for the evaluation of different design alternatives or for providing a customer quotation.
To support RPD, in the cost estimation process, a number of product and process requirements and constraints in the areas of part features, mould design, mould making processes, economics of mould making and part moulding costs, etc. should be considered simultaneously. Hence, the IDFC system is designed to be used in a concurrent engineering environment to provide board density information, scrap and rework cost estimates and a breakdown of set-up, labour and material costs for each step of the injection moulds development process. As shown in Figure 9.4, the IDFC system mainly includes the following entities:

1) User & Design interface: Interfaces can be used by design teams/designers through the Internet.

2) Costing algorithms: developed to perform cost calculation and estimation. (Muir 1998 and Xie et al 2001)

3) IDFC knowledge database: designed based on the breakdown costs and the detailed steps of mould development processes.

4) IDFC tools: designed to perform functions, e.g. gather and present facts, provide suggestions for possible cost reduction, diagnose costly designs, etc.

As shown in Figure 9.5, the IDFC system also contains:

1. An Internet based DFC data/knowledge base that contains specific guidelines organized in an appropriate form for easy access.

2. Internet based tools for cost calculation, data operations (e.g. edit, search and retrieve) of the DFC data/knowledge base and other databases.
3. DFC checklists and guidelines, which provide examples of good economical design, and remind the designer to check various costly designs. Product designers can easily get the knowledge of a product design and cost by checking the detailed cost information from the mould product id and feature ID.

The DFC guidelines can be automatically reorganized and reformulated according to the sub-components of a new mould (or product) and the generic DFX model. This will allow the manufacturing company to meet the dynamic changes in its global market by providing products with suitable prices.

To test the IDFC system with these cost functions, a case study has been carried out to determine the cost of making an injection mould, which is used to make the front and back panels of a home power manager as shown in Figure 9.6. In order to reduce the costs, the front and rear panels were molded together in a two cavity family mould. One of each is therefore produced with every injection cycle. The injection mould is shown in Figure 9.7. It is a two ways split injection mould with a ‘Z gate’ runner system. This runner system carries the molten polymer to the backside of the component so that the injection point blemishes end up on the inside of the final assembly and unseen by the users. The information in the pricing schedule makes it possible to form cost estimates for the various components of an average moulding tool.

As discussed in section 9.3, A DFC knowledge base was built to model the mould product development information and cost information. The pricing schedule of the moulding tool provided by the manufacturing companies is modelled in the DFC knowledge base. These cost information data in the DFC knowledge base are used for the cost estimation.

For the home power manager, the manufacturing costs can be estimated with the costing methods above, which are shown in Table 9.1. These values can be seen from the Web page in Figure 9.4. The mould cost generated by the IDFC software tool is NZ$32,316. In order to calculate the labour cost for machining EDM electrodes and the cavity with the equations created above, the relevant ratios for estimating machining costs have been identified. A more detailed report for the machining costs is presented by Muir (1998).

<table>
<thead>
<tr>
<th>Item</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Materials</td>
<td>$2308.00</td>
</tr>
<tr>
<td>Mould Base Cost</td>
<td>$1100.00</td>
</tr>
<tr>
<td>Heat Treatment</td>
<td>$200.00</td>
</tr>
<tr>
<td>Freight</td>
<td>$300.00</td>
</tr>
<tr>
<td>Labour @ $53/hr</td>
<td>$28408.00</td>
</tr>
<tr>
<td></td>
<td>$32316.00</td>
</tr>
</tbody>
</table>

Table 9.1. Costs for the manufacture of the injection mould
9.5 Chapter Conclusion

In this chapter, a new approach for developing the IDFX systems is introduced for rapidly and economically developing plastics injection moulds. The structures of the systems are presented. A generic DFX model is introduced for developing various IDFX systems. Based on this data model, DFX knowledge bases for plastic injection mould development have been built and Internet based DFX (IDFM and IDFC) systems have been built up. These tools utilize algorithms that predict the level of activity at each step in the manufacturing system, as driven by the decisions about the design and production of injection moulds. The IDFM/IDFC systems allow engineers to try design alternatives, with immediate feedback given, to help lead them to most cost effective and rapid designs. These systems have several important features. First, the IDFM and the IDFC systems can be used in concurrent engineering environments. Second, the developed IDFM and IDFC systems are based on a generic DFX model. Various software tools that are employed at different stages of the mould development can easily access this model without a middle data translator. Second, the management of the DFX data/knowledge base becomes relatively easy. Third, the DFX guidelines are organized in an appropriate form and can be automatically reorganized for modular products. Last, the IDFX system can be easily integrated with other Internet based product development software tools or systems.
CHAPTER 10
CONCLUSIONS AND FUTURE WORK

This last chapter will conclude the research results obtained from this PhD thesis work. It starts with a brief summary of the goal of this thesis work and the problems faced by this Ph.D. work. With respect to the thesis objectives as stated in Chapter 1, the major achievements or research findings are summarized in section 10.2 of this chapter. In the final section of this thesis, the possible future research directions and topics are discussed.

10.1 The Goal of This Thesis Work and the problems facing it

The primary goal of this thesis is to improve OKP companies’ rapid product development ability so that they can remain profitable in their globally and dynamically competitive market. In order to realize this goal, it is necessary to develop new OKP systems, strategies, methods and algorithms for OKP companies to properly handle these globally and dynamically competitive issues. The integration of new technology and methodologies under the global heterogeneous environment was the main problem of this thesis research. This problem was well addressed by an Internet based integrated product development system which was proposed in this thesis. To apply widely this product development system in OKP sheet metal part/product or tool/mould development, some modules, which are the important entities of the system, and some relevant methods, algorithms and concepts were also developed through this thesis work. These modules, methods, algorithms and concepts will be further summarized in the following section.

10.2 Major Achievements and Research Findings

The achievements of this PhD research thesis work mainly include a reference system architecture, methods, algorithms, concepts and the necessary computer software tools and interfaces for Internet based rapid OKP development. At the end of the thesis work, there are four software packages have been developed. The main findings from this thesis work have been published in international journals and conference proceedings as listed in the SUMMARY of this thesis. The source codes of the main software tools and interfaces have been attached in the Appendix, which include:

(1) An integrated software platform for integrating various engineering software systems on the Intra/Internets. This software platform was developed by using computer inter/intranet communication and interfacing technology. It was designed as a WINDOW based and user-friendly software system. It can be used to integrate all kinds of computer aided engineering software packages, e.g. CAD system, computer aided process planning (CAPP) system, cost analysis system, and manufacturing process simulation system, and shop floor operation simulation system, to effectively support an OKP company’s global product development practice. This software platform includes the relevant databases and knowledge bases as presented in this thesis.

(2) A computer aided product design, process planning and manufacturing software package for intelligent and economical product development.
(3) A WWW based software package for communication and data management on the Intranet/Internets during a collaborative product design and manufacturing process between an OKP company and its partners/customers.

The main research findings obtained or contributions made from this thesis work are summarized in the following:

1) Current rapid OKP product development systems, methods and technology review and comparison. An extensive literature review was given in Chapter 2 on the historical background of OKP and Internet based product design and manufacture systems. From this literature review, the current approaches on rapid OKP product development were evaluated and discussed against the needs for recently emerging global OKP product development practices. Hence the gaps and problems between the current technological state and the practical needs have been identified. These gaps can be filled and problems can be solved through developing an Internet based computer aided product development system and a global collaborative product development concept and method. Hence a reference system architecture for the Internet based rapid OKP product development system was developed and proposed at the end of Chapter 2 of this thesis.

2) The Internet based integrated product development system for rapidly developing OKP sheet metal part/product

This is in fact an application of the proposed reference system architecture to the rapid development of OKP sheet metal parts/products. From this successful industrial test and implementation, the feasibility of the proposed reference system architecture has been proved. However, to meet some special needs for sheet metal part/product development, some technical modules and computer interface were developed as the entities of the system. These include structure, a RTCAPP (real time computer aided process planning) module, customer interfaces, a simulation platform and a design/manufacturing knowledge bases module for supporting concurrent product design and manufacturing. The definition and the structure of the integrated data structure for concurrent sheet metal part/product design and manufacturing was also given in Chapter 3 of this thesis. This Internet based rapid OKP sheet metal part/product development system has been tested with several industrial cases. These industrial implementations clearly show that this Internet based OKP sheet metal part/product development system has a lot of advantages over the traditional systems.

3) An Information Modelling Framework for Intelligent and Concurrent Sheet Metal Parts Design and Manufacturing

As information integration is an important issue in supporting integrated and concurrent product development, i.e. rapid product development, the structure of an information framework for concurrent design and manufacturing of OKP sheet metal parts/products were developed and presented in Chapter 4 of this thesis. This information framework was developed to build an information bridge to fill the electronic data communication gaps among the sheet metal part/product design system, process planning system and manufacturing/production system. The principles of zero thickness and zero bend radius were proposed, which can be used to abstract the geometry entities of sheet metal parts in order to facilitate part modelling and information modelling. A tree based information modelling methodology was also presented in Chapter 4 as the important concept and method to support the implementation of this information framework. This information framework enhances the overall system integration of the Internet based rapid OKP sheet metal part/product development system. Its feasibility was tested by an industrial case study as reported at the end of Chapter 4.

4) Compound OKP sheet metal part/product manufacturing process.
To speed up an OKP sheet metal part/product development, a compound cutting and punching method was developed and tested on an industrial case. This compound sheet metal manufacturing process can simplify the sheet metal cutting and punching process planning and hence simplify the part/product design. This simplification can significantly reduce a sheet metal part development cycle time. To implement this compound sheet metal manufacturing process, an integrated CAD/CAPP/CAM system was developed and presented in Chapter 5 owing to the fact that existing commercial CAD/CAM systems are not suitable for this manufacturing method, especially under concurrent and global product development strategy. Some problems have to be solved before these existing commercial CAD/CAM systems can be employed and integrated for this compound manufacturing method. Some solutions to solve these problems were suggested. These include an integrated data exchange platform developed on the Pro/INTRALINK of the ProEngineer (a commercial CAD/CAM package) and in the format of STEP, and a knowledge-based real time CAPP (RTCAPP) system for compound sheet metal cutting and punching. In addition to these, some modules were also developed as the entities of the integrated CAD/CAPP/CAM system. They include an automatic tool selection and manufacturing process planning module, a shortest tool path optimisation module, and an automatic insertion of auxiliary path module based on knowledge bases.

5) Cost estimation and optimisation for rapid development of sheet metal part/product

To be able to provide in-time cost estimates and optimisation results to support the product developers’ decision-making in RPD, a feasible computer aided automatic cost estimation and optimisation system was developed. This system was built on a Cost Index Structure, which is generic and can be applied to all kinds of products development. The cost estimate methods and optimisation algorithms are effective. They can be easily understood by and implemented in industry. This concurrent approach enables product developers to design the product, plan the processes, schedule operations on the shop floors, manage logistics, and estimate and optimally control the cost feature by feature.

6) Integrated product data model for product development life cycle

An integrated product data model for product development life cycle was developed. The emphasis of this research work was placed on the integrated data management and knowledge reuse to support a company’s practice to shorten its product development cycle. The integrated data structure was modelled through using the EXPRESS of STEP. In terms of this data structure, a design/manufacturing knowledge base was developed as a major part of the WWW based product development system. This data structure was successfully applied to supporting a rapid injection mould development.

7) Internet-based Product Information Management System for Rapid OKP Product Development

To facilitate the complicated data exchange and communications between a master OKP company and its branches or partners via Intra/Internets, an internet-based product information management system was developed. This information management system includes 1) WWW based, distributed and object oriented databases and knowledge bases for product design and manufacturing; 2) A WWW based database management system; 3) Web browser interface to link a local database to a web page browser; and 4) An integrated software platform for the integration of CAD, CAPP, CAM and WWW based information management systems. This Internet based information management system was implemented in a New Zealand manufacturing company, which has its headquarter in Christchurch and a number of branches or partners in other places in New Zealand and overseas.

8) Internet Based DFX for Rapid OKP Product Development
Chapter 10 Conclusion and Future Work

The application of Design for X (DFX), which is an effective method in concurrent engineering for RPD, to the global and collaborative OKP product development has been tested in this thesis work. An Internet based DFX or IDFX system was developed for rapid tool/mould making. The structures of the system were presented and discussed in Chapter 9. A generic IDFX system structure was further introduced in Chapter 9 for developing various IDFX systems. Based on this system structure, IDFX knowledge bases for a plastic injection mould development have been created and the Internet based DFX (i.e. internet based design for manufacturing (IDFM) and internet based design for cost (IDFC)) systems have been built up. The IDFM/IDFC systems allow engineers to evaluate alternative designs, with immediate feedbacks on manufacturability and cost and thus to help them to quickly come to an effective and economical design.

10.3 Future Work

As an increasing level of customisation and the demand for individualized products has become a clear trend in the international manufacturing market, more and more manufacturing companies are being forced to move towards OKP and develop OKP products at a faster and faster pace. Hence, the Internet based rapid OKP product development as studied in this thesis work would become an important research area in the development of a new generation of manufacturing enterprises and systems.

It can be seen from this thesis that the development of suitable systems, methods and technology for rapid OKP product development in a global and collaborative environment is a large and extensive research task. Although a lot of research efforts have been made through this thesis work, there are still quite a few problems unsolved. The research work as reported in this thesis can be further extended and some possible research directions have been identified, which are summarized in the following:

Extending the Internet based rapid OKP Product development system

In chapter 2, the structure and major components of the Internet based rapid OKP product development system have been defined. However, for the development of different OKP products, some of the components may be different. As illustrated in chapter 3, the product development system for sheet metal part/product development contains some specific modules such as nesting, process planning and costing. Hence, to be able to use the integrated product development system for rapidly developing other OKP products, further research should be carried out on the compatibility and interoperability of the proposed reference system architecture.

Continually upgrading the Internet based rapid OKP product development system

As the product development system depends on new technology and theories, the proposed reference system architecture has to be continually upgraded with the development of modern technology. The new technology may affect data standards and modelling methodologies, knowledge modelling methods and AI support; global optimisation algorithms, and Internet communication, etc. Only by keeping this continually upgraded, can the integrated OKP product development system thus keep its advantage over other systems, and thus help companies meet the increasing market competition.

Finding New Implementing Methodologies

The Internet technology sounds as if it can be used by most global manufacturing companies. Quite a few research efforts as reviewed in chapter 2 as well as the research work as reported in this thesis, have been made to develop various kinds of RPD systems. However, it is still not clear how effectively these developed systems and methods can be implemented to support rapid OKP product development in a global and collaborative environment. Particularly some software interfacing and developing tools and methods are
needed to efficiently develop and effectively implement this type of Internet based product development system. These tools and methods will be able to provide a good connection between modern computer technology and a rapid OKP product development process. They can be used to decrease the time consumed on the programming work and shorten the time of updating an existing Internet based rapid OKP product development system. This research requires cooperation from both computer science and manufacturing areas.

Achieving High Level System Integration

As IRPD requires the integration of people, business processes and information technology through a whole product development cycle, this requires a high level system integration across all the organisational functions in an OKP company as well as its suppliers and customers. The information of a product on its design, manufacturing, use, maintenance and disposal is not only shared between the departments within a company, but also shared between the “partner” companies in the world (Wagner et al. 1996). This wider information sharing practice in today’s manufacturing requires the systems integration that covers not only the information sharing through a central database in tradition, but also the technologies for product design, process planning, computer simulation, manufacturing, knowledge reuse and optimisation algorithms applied in a product development. To meet this demand of high level system integration, the new data structures, models and software developing tools need to be developed, such as: (1) Life-time product data structure and models for integrating various computer aided engineering systems in a product development cycle; (Tu et al. 2001); (2) Definition of generic attributes and consistent data models for PD process; (3) The further development of STEP and improving its standardization and system interoperability.

Developing new CSCW and interfacing techniques

Since an IRPD system requires the cooperation of various engineering systems over a network environment, the CSCW and interfacing techniques are continually important for developing the IRPD systems. They directly affect the accuracy and efficiency of the data management through a whole PD cycle. However, early and even recent experiences have shown that the CSCW technologies have not been so successfully applied in the development of Internet based product development systems (Luchich 1995). The critical problems mainly result from the limitation of the technology itself and the compatibility with other technical systems and social norms.

On the other hand, the interfacing techniques in the IRPD systems should have the following capabilities: 1) Workbench (common users interface to multiple PD applications); 2) Process modelling and workflow management; 3) Application encapsulation and invocation; and 4) Application data translation and transfer and multi-vender networking and platform support. The common used data exchange standards, e.g. CAD-NT, CGI, CGM, SET, PDES, STEP, and IGES, do not have the functionalities to directly support interfacing via variant application software packages. Although great efforts have been made through this thesis work and also by many other researchers, e.g. Wang et al. (1995), Lau and Jiang (1998); Dorador and Young (1999), Tu et al. (2000c and 2001), to resolve this problem, the research of CSCW and interfacing techniques for the purpose of RPD is still in a earlier developing stage. Further research is still needed, such as by using STEP to represent non-geometric information (Wang et al. 1995), the combination of the Internet modelling methodology (e.g. HTML, CORBA, etc.) with STEP (Hardwick, et al. 1996), and the Internet based data translators between various data formats (Zhang et al. 2000). These would be the interesting research topics for IRPD.
Expanding the information integration framework

As the important technology to improve the system integration, the information integration framework as proposed in chapter 4 was successfully used to fill the information gap between the design and other downstream sheet metal part/product development processes. Although this information integration framework was developed for rapid OKP sheet metal part/product development, in my belief, it can be further developed as a generic information integration framework for other OKP product development.

Improving the cost estimating and optimisation algorithm

From the foregoing discussion, it is clear that cost estimation is an area of great research interest. New technologies such as neural networks (Ehrlenspiel and Schaal 1992, Bode 1998 and 2000), activity based costing (Tamas et al. 2000), log-linear and non-linear learning curve model (Timothy 1999) AI and mathematical models (Xie et al. 2001a) have been widely researched. However, a widely accepted costing system or a system with a wide applicability has not been found. Chapter 6 proposed a cost estimate and control methodology for rapid sheet metal part/product development. The cost optimisation algorithm as presented in Chapter 6 was based on a multi-objective algorithm with the support of cost resources. This algorithm works well when all the needed information is ready. But at the very early design stage, as there is not enough design or downstream information available for the estimation algorithm. New and intelligent cost estimation and optimisation algorithms need to be further researched.

Further development of Internet based DFX

In chapter 9, a prototype of IDFX system was introduced for rapid tool/mould making. To further apply this prototype IDFX system for rapidly developing other types of OKP products, some research efforts are needed, such as 1) The detailed DFX guidelines should be collected for various functions of "X" and compiled in the IDFX systems; 2) The standard interfaces of the IDFX systems with product designers/design team should be created for the purpose of providing right information at the first time; 3) The IDFX systems need to provide tools for building new DFX functions and criteria for setting up the relevant guidelines; and 4) Further researches for generic DFX model for developing various IDFX systems are needed.
## Appendix 1: Some of the Internet based PD systems

<table>
<thead>
<tr>
<th>Some Internet based PD products &amp; Research Projects</th>
<th>Purpose</th>
<th>Data Formats and compatible softwares</th>
<th>Key implementation Tech.</th>
<th>Stage of the product development process</th>
<th>Developer</th>
<th>Key Functionalities</th>
</tr>
</thead>
<tbody>
<tr>
<td>WinCHILL</td>
<td>Life cycle management</td>
<td>Compatible with CAD softwares, e.g. AutoCAD, Pro/ENGINEER. Support standard data formats, e.g. STEP, IGES, DXF and VRML and Internet technologies, e.g. CORBA, OLE.</td>
<td>RAD Tool, JAVA platform</td>
<td>Design, Process planning and Manufacturing</td>
<td>WinCHILL Corporation</td>
<td>Product definition Product Planning Customer Management Maintenance &amp; Repair Production Manufacturing Planning Sourcing Product Modelling</td>
</tr>
<tr>
<td>SMARTTEAM</td>
<td>Product data management</td>
<td>Connect with CAD, MRP through API function</td>
<td>Smart Basic/ Visual Basic</td>
<td>Product Design and planning.</td>
<td>Smart Solution</td>
<td>Document management and workflow management</td>
</tr>
<tr>
<td>QUEST</td>
<td>Process simulation and optimization</td>
<td>Provides standard interface with CAD and supports IGES</td>
<td>Similar with Pascal programing language</td>
<td>Process planning &amp; optimization</td>
<td>Denhe Robotics</td>
<td>Virtual Simulation</td>
</tr>
<tr>
<td>CVW (Collaborative Virtual Workspace)</td>
<td>Communicating with partners</td>
<td>Internet, Video conference software</td>
<td>Java</td>
<td>Earlier design stage</td>
<td>The MITRE Corporation</td>
<td>Audio/video conferencing Conference servers Text chat &amp; Instant Messaging Data conferencing Place based collaboration environments</td>
</tr>
<tr>
<td>CIMOSA project</td>
<td>CIM Open Structure</td>
<td>Internet, enterprise object, process modelling</td>
<td>Enterprise modelling and integration</td>
<td>The whole PD processes</td>
<td><a href="http://cimosa.cn">http://cimosa.cn</a> t.pl/</td>
<td>Enterprise integration, product management Enterprise structure Enterprise engineering</td>
</tr>
<tr>
<td>FixtureNet</td>
<td>Fixture design</td>
<td>Interactive CAD via the WWW</td>
<td>HTML, Java</td>
<td>Design</td>
<td>Design stage</td>
<td></td>
</tr>
<tr>
<td>OPDX</td>
<td>Collaboration design and information sharing to develop</td>
<td>PD with sharing information</td>
<td>Web Collaboration Tools,</td>
<td>Partner tools Product Information management Workflow</td>
<td>ORACLE corporation</td>
<td>Design and product data management</td>
</tr>
<tr>
<td>Appendix 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Competitive Products</strong></td>
<td><strong>Document Management</strong></td>
<td><strong>Additional Information</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testbeds</td>
<td>Collaborative research</td>
<td>STEP</td>
<td>Visualization tools, STEP, information modeling, etc.</td>
<td>Test platform</td>
<td>NIST</td>
<td>Collaborative design, manufacturing, process planning, etc.</td>
</tr>
<tr>
<td>METEOR2</td>
<td>Workflow management</td>
<td>Internet based workflow management</td>
<td>CORBA, HTML, JAVASCRIPT, and CGI</td>
<td>Design and planning</td>
<td><a href="http://www.scara.org/hit.html">http://www.scara.org/hit.html</a></td>
<td>Internet based Planning and management</td>
</tr>
<tr>
<td>CyberSystem</td>
<td>Supply chain planning and enterprise resource planning</td>
<td>ERP systems and supply chain systems</td>
<td>Not mentioned</td>
<td>Seamless integration of supply chain planning and enterprise resource planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProcessPoint</td>
<td>Collaborative design</td>
<td>Design and planning stage</td>
<td><a href="http://www.processpoint.com">http://www.processpoint.com</a></td>
<td>Collaborative design, deployment and manufacturing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFMC</td>
<td>Workflow management</td>
<td>PDM, knowledge management</td>
<td>Planning stage</td>
<td><a href="http://www.wfmc.org">http://www.wfmc.org</a></td>
<td>Internet based WFM Process management and optimization</td>
<td></td>
</tr>
<tr>
<td>ISCM</td>
<td>Supply chain management</td>
<td>Supplier’s resource management systems</td>
<td>Agent technology</td>
<td>Supply management</td>
<td>Integration laboratory University of Toronto</td>
<td>Information agents Temporal reasoning Scheduling Resource management</td>
</tr>
<tr>
<td>ABMA</td>
<td>Enterprise Integration</td>
<td></td>
<td>Budenske et al. 1998 Architecture Tech. Co.</td>
<td>Middleware architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AARIA</td>
<td>Manufacturing scheduling and control</td>
<td></td>
<td>Paramarak et al. 1998</td>
<td>Using autonomous agents to represent physical entities, processes and operations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IA</td>
<td>Enterprise</td>
<td></td>
<td>Pan and</td>
<td>Using large number</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Appendix 1

<table>
<thead>
<tr>
<th>framework</th>
<th>Integration</th>
<th>Tenenbaum, 1991</th>
<th>of intelligent agents for integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAO</td>
<td>Intelligent Manufacturing</td>
<td>Kwok and Norris, 1994 U. of Calgary</td>
<td>A rule-based object system for developing intelligent manufacturing software</td>
</tr>
<tr>
<td>iDCSS</td>
<td>Concurrent Engineering</td>
<td>Klein 1995</td>
<td>Propose an integrated model to combine existing coordination technologies</td>
</tr>
<tr>
<td>MetaMorph II</td>
<td>Intelligent manufacturing and supply chain management</td>
<td>Shen et al 1998, University of Calgary</td>
<td></td>
</tr>
<tr>
<td>Madefast</td>
<td>Collaborative Engineering</td>
<td>Cutkosky et al 1996, Stanford</td>
<td>Using Internet as a platform for collaboration</td>
</tr>
<tr>
<td>AMC</td>
<td>Manufacturing Scheduling</td>
<td>Goldsmith and Interrante 1998</td>
<td>Using physical agents: part agents and machine agents</td>
</tr>
<tr>
<td>CIIMPLEX</td>
<td>Enterprise Integration</td>
<td>Peng et al 1998 UMBC EE/COMS</td>
<td>Service agents (Name Server, Facilitator agent, Gateway Agent)</td>
</tr>
<tr>
<td>ADE</td>
<td>Supply Chain Management</td>
<td>Mehra and Nissen 1998</td>
<td>Using delegation-based handling similar to JavaBeans</td>
</tr>
<tr>
<td>VPD</td>
<td>Virtual product development</td>
<td>The whole PD process</td>
<td>NIST Redefining Product Development Virtual Enterprise Service Centre Virtual Enterprise Architecture</td>
</tr>
</tbody>
</table>
Appendix 2: QFD Program

Part 1: Programs for QED Interface and Interactions with Microsoft Access Database

Private Sub BoxA_Click()
End Sub

Private Sub cmdClearAll1_Click()
BoxA.Clear
End Sub

Private Sub cmdClearAll2_Click()
BoxB.Clear
End Sub

Private Sub cmdDeleteA_Click()
BoxA.RemoveItem BoxA.ListIndex
End Sub

Private Sub cmdDeleteB_Click()
BoxB.RemoveItem BoxB.ListIndex
End Sub

Private Sub cmdEnter_Click()

Dim iRow As Integer
Dim iNumRow As Integer

Application.ScreenUpdating = False

ListSizeA = BoxA.ListCount
For iRow1 = 1 To ListSizeA
  With Worksheets("QFD Table")
    .Range("B3").Cells(iRow1, 1).Value = BoxA.List(iRow1 - 1)
    .Activate
  End With
Next iRow1

ListSizeB = BoxB.ListCount
For iRow2 = 1 To ListSizeB
  With Worksheets("QFD Table")
    .Range("D2").Cells(1, iRow2).Value = BoxB.List(iRow2 - 1)
  End With
Next iRow2

For iRow2 = 0 To ListSizeB
  With Worksheets("QFD Table")
    .Range("C2").Cells(1, iRow2 + 1).Borders(xlEdgeTop).Weight = xlMedium
    .Range("C3").Cells(1, iRow2 + 1).Borders(xlEdgeTop).Weight = xlMedium
    .Range("A3:B3").Borders(xlEdgeTop).Weight = xlMedium
  End With
Next iRow2
For iRow1 = 1 To 34
For iRow2 = 0 To ListSizeB + 1
For I Bor = -1 To 0
    With Worksheets("QFD Table").Range("D3")
        .Cells(iRow1, iRow2 - 1).Borders(xlEdgeRight).Weight = xlThin
        .Cells(iRow1, iRow2 - 1).Borders(xlEdgeBottom).Weight = xlThin
        .Cells(iRow1, 0).Borders(xlEdgeBottom).Weight = xlThin
        .Cells(1, I Bor).Borders(xlEdgeTop).Weight = xlMedium
        .Cells(0, iRow2).Borders(xlEdgeLeft).Weight = xlThin
    End With
Next I Bor
Next iRow2
Next iRow1

For iRow2 = 1 To ListSizeB
    With Worksheets("QFD Table").Range("D1")
        .Cells(1, iRow2).Borders(xlEdgeTop).Weight = xlMedium
    End With
Next iRow2

For iRow2 = -2 To ListSizeB
    For INumRow = -1 To ListSizeB
        With Worksheets("QFD Table")
            .Cells(33, iRow2 + 3).Borders(xlEdgeBottom).Weight = xlMedium
            .Cells(36, INumRow + 3).Borders(xlEdgeBottom).Weight = xlMedium
        End With
    Next INumRow
Next iRow2

sBorderLeft = Cells(3, 1).Address & ":" & Cells(33, 1).Address
sBorderCenter0 = Cells(3, 1).Address & ":" & Cells(36, 1).Address
sBorderCenter1 = Cells(1, 3).Address & ":" & Cells(36, 3).Address
sBorderRight = Cells(1, BoxSizeB + 3).Address & ":" & Cells(36, 3 + ListSizeB).Address
With Worksheets("QFD Table")
    .Range(sBorderLeft).Borders(xlEdgeLeft).Weight = xlMedium
    .Range(sBorderCenter0).Borders(xlEdgeRight).Weight = xlMedium
    .Range(sBorderCenter1).Borders(xlEdgeRight).Weight = xlMedium
    .Range(sBorderRight).Borders(xlEdgeRight).Weight = xlMedium
End With

"Merge cells"

sMerge1 = Cells(3, 1).Address & ":" & Cells(33, 1).Address
sMerge2 = Cells(1, 4).Address & ":" & Cells(1, 3 + ListSizeB).Address
With Worksheets("QFD Table")
    .Range(sMerge1).Merge
    .Range(sMerge1).Value = "Voice of Customer"
    .Range(sMerge2).Merge
    .Range(sMerge2).Value = "Voice of Engineer"
    .Cells(2, 3).Value = "Customer Importance"
    .Cells(34, 2).Value = "Absolute Importance"
    .Cells(35, 2).Value = "Customer Satisfaction Performance"
    .Cells(36, 2).Value = "Competitive Benchmark"
    .Cells(38, 2).Value = "*Note: All information are based"
Appendix 2

.Cells(39, 2).Value = "On product code PA337D"
.Cells(41, 2).Value = "Note: All values for Competitive"
.Cells(42, 2).Value = "Benchmark based on 1-5 Rating"
.Cells(43, 2).Value = "Customer Performance are based"
.Cells(44, 2).Value = "on 1-9 rating"
.Cells(46, 2).Value = "Note: All values in Relation Matrix"
.Cells(47, 2).Value = "are of <> 1, 3 & 9 rating"

End With
ActiveSheet.Columns("A:A").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 90
    .ShrinkToFit = False
End With
ActiveSheet.Columns("A:B").Select
With Selection
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0
    .ShrinkToFit = False
End With
ActiveSheet.Rows("2:2").Select
Selection.Orientation = 90
Selection.Rows.AutoFit
Selection.HorizontalAlignment = xlCenter
Selection.VerticalAlignment = xlBottom
ActiveSheet.Columns("B:B").Select
Selection.Columns.AutoFit
ActiveSheet.Range("A1").Select

' Technical Correlation

ListSizeB = BoxB.ListCount
For iRow2 = 1 To ListSizeB
    With Worksheets("Tech Correlation")
        .Range("B3").Cells(iRow2, 1).Value = BoxB.List(iRow2 - 1)
        .Range("C2").Cells(1, iRow2).Value = BoxB.List(iRow2 - 1)
    .Activate
    End With
Next iRow2

sHighlight = Cells(2, 4).Address & ":" & Cells(2, 3 + ListSizeB).Address
sColumn = Cells(1, 2).Address & ":" & Cells(60, 3 + ListSizeB).Address
sRelation = Cells(2, 2).Address & ":" & Cells(2 + ListSizeB, ListSizeB + 2).Address
sBorder1 = Cells(2, 2).Address & ":" & Cells(2 + ListSizeB, 2).Address
sBorder2 = Cells(2, 2).Address & ":" & Cells(2, 2 + ListSizeB).Address

ActiveSheet.Range(sHighlight).Select
Selection.Copy
ActiveSheet.Range(sColumn).EntireColumn.AutoFit
Application.CutCopyMode = False
With Selection
  .HorizontalAlignment = xlLeft
  .VerticalAlignment = xlBottom
  .WrapText = False
  .Orientation = 0
  .IndentLevel = 0
  .ShrinkToFit = False
  .MergeCells = False
End With

ActiveSheet.Rows("2:2").Select
Selection.Orientation = 90
Selection.Rows.Adf
Selection.HorizontalAlignment = xlCenter
Selection.VerticalAlignment = xlBottom
ActiveSheet.Columns("B:B").Select
Selection.Columns.Adf
ActiveSheet.Range("A1").Select

' Technical Correlation's Borders

ActiveSheet.Range(xlRelation).Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
  .LineStyle = xlContinuous
  .Weight = xlThin
  .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
  .LineStyle = xlContinuous
  .Weight = xlThin
  .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
  .LineStyle = xlContinuous
  .Weight = xlThin
  .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
  .LineStyle = xlContinuous
  .Weight = xlThin
  .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
  .LineStyle = xlContinuous
  .Weight = xlThin
  .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With

Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
Sheet.Range(sBorder1).Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
Appendix 2


End With With Selection.Borders(xlEdgeTop)

End With With Selection.Borders(xlEdgeBottom)

End With With Selection.Borders(xlEdgeRight)

End With With Selection.Borders(xlInsideHorizontal)

End With With Selection.Borders(xlInsideVertical)

End With With Selection.Borders(xlOutsideHorizontal)

End With With Selection.Borders(xlOutsideVertical)

End With

'Merge Technical Correlation

sMerge3 = Cells(1, 3).Address & ":" & Cells(1, 2 + ListSizeB).Address ActiveSheet.Range(sMerge3).Select With Selection
Appendix 2

Private Sub image1_Click()
End Sub

Private Sub InsertA_Click()
    BoxA.AddItem txtBoxA.Text
    txtBoxA.Text = ""
End Sub

Private Sub InsertB_Click()
    BoxB.AddItem txtBoxB.Text
    txtBoxB.Text = ""
End Sub

Private Sub lblCustomer_Click()
End Sub

Private Sub lblEngineer_Click()
End Sub

Private Sub BoxB_Click()
End Sub

Private Sub txtBoxA_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    If KeyCode = vbKeyReturn Then
        If txtBoxA.Text <> "" Then
            BoxA.AddItem txtBoxA.Text
            txtBoxA.Text = ""
        End If
    End If
End Sub

Private Sub txtBoxB_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    If KeyCode = vbKeyReturn Then
        If txtBoxB.Text <> "" Then
            BoxB.AddItem txtBoxB.Text
            txtBoxB.Text = ""
        End If
    End If
End If
End Sub

Private Sub txtTitle_Change()

End Sub
Part2: Programs for QFD Calculations and Table Operations

Private Sub cmdCalculate_Click()
ListSizeB = Worksheets("Input").BoxB.ListCount
sCalc = Cells(34, 4).Address & "." & Cells(34, ListSizeB + 3).Address
sEndCalc = Cells(34, ListSizeB + 3).Address
ActiveSheet.Range("D34").Select
ActiveCell.FormulaR1C1 = 
Selection.AutoFill Destination:=Range(sCalc), Type:=xlFillDefault
ActiveSheet.Range(sCalc).Select
With Selection.Borders(xlInsideVertical)
 .LineStyle = xlContinuous
 .Weight = xlThin
 .ColorIndex = xlAutomatic
End With
ActiveSheet.Range(sEndCalc).Borders(xlEdgeRight).Weight = xlMedium
ActiveSheet.Range("B34").Select
End Sub

Private Sub cmdClear_Click()
Cells.ClearSelect
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Selection.ColumnWidth = 4
Selection.RowHeight = 13.2
With Selection
 .HorizontalAlignment = xlCenter
 .VerticalAlignment = xlCenter
 .Orientation = 0
 .WrapText = False
 .ShrinkToFit = False
 .MergeCells = False
End With
Selection.ClearContents
Range("F15").Select
With Worksheets("Input")
 .Activate
End With
End Sub

Private Sub cmdGraph_Click()
Appendix 2

ListSizeB = Worksheets("Input").BoxB.ListCount
sPlotvalue = Cells(34, 4).Address & ";" & Cells(34, 3 + ListSizeB).Address
sXvalue = Cells(2, 4).Address & ";" & Cells(2, 3 + ListSizeB).Address

ActiveWindow.ScrollRow = 1
Range(sPlotvalue).Select
Charts.Add
ActiveChart.ChartType = xl3DColumnClustered
ActiveChart.SetSourceData Source:=Sheets("QFD Table").Range(sPlotvalue), PlotBy:=xlColumns_
ActiveChart.SeriesCollection(1).XValues = Worksheets("QFD Table").Range(sXvalue)
ActiveChart.Location Where:=xlLocationAsNewSheet, Name:="QFD Chart"
ActiveChart.Axes(xlCategory).Select
Selection.TickLabels.Orientation = xlUpward
With ActiveChart
    .RightAngleAxes = True
    .HasTitle = True
    .ChartTitle.Characters.Text = "Absolute Importance for Engineer"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Rating"
    .Axes(xlValue, xlPrimary).AxisTitle.Orientation = 90
End With
With ActiveChart.Axes(xlCategory)
    .HasMajorGridlines = False
    .HasMinorGridlines = False
End With
With ActiveChart.Axes(xlValue)
    .HasMajorGridlines = False
    .HasMinorGridlines = False
End With
ActiveChart.HasLegend = False
ActiveChart.PlotArea.Select
Selection.Top = 31
Selection.Height = 387
With Selection.Border
    .ColorIndex = 30
    .Weight = xlThin
    .LineStyle = xlContinuous
End With
Selection.Interior.ColorIndex = 19

ActiveChart.SeriesCollection(1).Select
With Selection.Border
    .Weight = xlThin
    .LineStyle = xlAutomatic
End With

Selection.Shadow = True
Selection.InvertIfNegative = False
With Selection.Interior
    .ColorIndex = 30
    .Pattern = 13
End With
Sheets("QFD Chart").Select
Sheets("QFD Chart").Move After:=Sheets("QFD Table")
Private Sub cmdGraph2_Click()
    ListSizeB = Worksheets("Input").BoxB.ListCount
    sPlotvalue = Cells(35, 4).Address & ";" & Cells(36, 3 + ListSizeB).Address
    sXvalue = Cells(2, 4).Address & ";" & Cells(2, 3 + ListSizeB).Address

    ActiveWindow.ScrollRow = 1
    Range(sPlotvalue).Select
    Charts.Add
    ActiveChart.ChartType = xl3DColumnClustered
    ActiveChart.SetSourceData Source:=Sheets("QFD Table").Range(sPlotvalue), PlotBy:=xlColumns,
    ActiveChart.SeriesCollection(1).XValues = Worksheets("QFD Table").Range(sXvalue)
    ActiveChart.Location WHERE:=xlLocationAsNewSheet, Name:="QFD Chart2"
    ActiveChart.Axes(xlCategory).Select
    With ActiveChart.TickLabels.Orientation = xlUpward
    With ActiveChart
        .RightAngleAxes = True
        .HasTitle = True
        .ChartTitle.Characters.Text = "Customer Satisfaction Performance"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Orientation = 90
    End With
    With ActiveChart.Axes(xlCategory)
        .HasMajorGridlines = False
        .HasMinorGridlines = False
    End With
    With ActiveChart.Axes(xlValue)
        .HasMajorGridlines = False
        .HasMinorGridlines = False
    End With
    ActiveChart.HasLegend = False
    ActiveChart.PlotArea.Select
    Selection.Top = 31
    Selection.Height = 387
    With Selection.Border
        .ColorIndex = 30
        .Weight = xlThin
        .LineStyle = xlContinuous
    End With
    Selection.Interior.ColorIndex = 19
    ActiveChart.SeriesCollection(1).Select
    With Selection.Border
        .Weight = xlThin
        .LineStyle = xlAutomatic
    End With
    Selection.Shadow = False
    Selection.InvertIfNegative = False
    With Selection.Interior
        .ColorIndex = 30
        .Pattern = xlSolid
    End With
Appendix 2

Private Sub CmdCalculate1_Click()
    ListSizeB = Worksheets("Input").BoxB.ListCount
    sCalc = Cells(35, 4).Address & ";" & Cells(35, ListSizeB + 3).Address
    sEndCalc = Cells(35, ListSizeB + 3).Address
    ActiveSheet.Range("D35").Select
    ActiveCell.FormulaR1C1 = 
        
            "={R3C3*R[-32]*C*R[-31]*C+R5C3*R[-30]*C+R6C3*R[-29]*C+R7C3*R[-28]*C+R8C3*R[-
            22]*C+R14C3*R[-21]*C+R15C3*R[-20]*C+R16C3*R[-19]*C+R17C3*R[-18]*C+R18C3*R[-
            17]*C+R19C3*R[-16]*C+R20C3*R[-15]*C+R21C3*R[-14]*C+R22C3*R[-13]*C+R23C3*R[-
            12]*C+R24C3*R[-
            5]*C+R31C3*R[-4]*C+R32C3*R[-3]*C+R33C3*R[-2]*C+R34C3*R[-1]*C+R35C3*R[-
            2]*C}"
    Selection.AutoFill Destination:=Range(sCalc), Type:=xlFillDefault
    ActiveSheet.Range(sCalc).Select
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    ActiveSheet.Range(sEndCalc).Borders(xlEdgeRight).Weight = xlMedium
    ActiveSheet.Range("B35").Select
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
End Sub
Part3: Programs for Tech Correlation

Private Sub cmdClear_Click()
  Cells.Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlNone
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    Selection.Borders(xlEdgeBottom).LineStyle = xlNone
    Selection.Borders(xlEdgeRight).LineStyle = xlNone
    Selection.InsideVertical.LineStyle = xlNone
    Selection.InsideHorizontal.LineStyle = xlNone
    Selection.ColumnWidth = 4
    Selection.RowHeight = 13.2
  With Selection
    .MergeCells = False
  End With
  Selection.ClearContents
  Range("A1").Select
  With Worksheets("Input")
    .Activate
  End With
End Sub
Appendix 3: Database Program

Part 1: Product database operation

Option Compare Database
Option Explicit

Private Sub Mould_Flow_Results_Click()
On Error GoTo Err_Mould_Flow_Results_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fPU Product Mold Flow Results"
    stLinkCriteria = "[Cat #]=" & "" & Me![Cat #] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Mould_Flow_Results_Click:
    Exit Sub

Err_Mould_Flow_Results_Click:
    MsgBox Err.Description
    Resume Exit_Mould_Flow_Results_Click

End Sub

Private Sub Project_Appraisal_Click()
On Error GoTo Err_Project_Appraisal_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "ISF Project Appraisals"
    stLinkCriteria = "[PA number]=" & "" & Me![Cat #] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Project_Appraisal_Click:
    Exit Sub

Err_Project_Appraisal_Click:
    MsgBox Err.Description
    Resume Exit_Project_Appraisal_Click

End Sub

Private Sub Product_Design_Questions_Click()
On Error GoTo Err_Product_Design_Questions_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fPU Product Design Questions"
    stLinkCriteria = "[cat #]=" & "" & Me![Cat #] & ""
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Appendix 3

Exit_Product_Design_Questions_Click:
    Exit Sub

Err_Product_Design_Questions_Click:
    MsgBox Err.Description
    Resume Exit_Product_Design_Questions_Click

End Sub
Private Sub Product_Designers_Click()
    On Error GoTo Err_Product_Designers_Click

        Dim stDocName As String
        Dim stLinkCriteria As String

        stDocName = "iPU Product Designers"
        stLinkCriteria = "[cat #]=" & "" & Me![Cat #] & ""
        DoCmd.OpenForm stDocName, , stLinkCriteria

    Exit_Product_Designers_Click:
        Exit Sub

Err_Product_Designers_Click:
    MsgBox Err.Description
    Resume Exit_Product_Designers_Click

End Sub
Private Sub Product_Assembly_Picture_Click()
    On Error GoTo Err_Product_Assembly_Picture_Click

        Dim stDocName As String
        Dim stLinkCriteria As String

        stDocName = "f Product assembly picture"
        stLinkCriteria = "[cat #]=" & "" & Me![Cat #] & ""
        DoCmd.OpenForm stDocName, , stLinkCriteria

    Exit_Product_Assembly_Picture_Click:
        Exit Sub

Err_Product_Assembly_Picture_Click:
    MsgBox Err.Description
    Resume Exit_Product_Assembly_Picture_Click

End Sub
Private Sub Tool_Make_Product_Click()
    On Error GoTo Err_Tool_Make_Product_Click

        Dim stDocName As String
        Dim stLinkCriteria As String

        stDocName = "fSF (Tools) Product Tool Relationship"
        stLinkCriteria = "[Cat #]=" & "" & Me![Cat #] & ""

    Exit_Tool_Make_Product_Click:
        Exit Sub
Appendix 3

DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Tool_Make_Product_Click:
    Exit Sub

Err_Tool_Make_Product_Click:
    MsgBox Err.Description
    Resume Exit_Tool_Make_Product_Click

End Sub
Private Sub Tool_Make_Part_Click()
On Error GoTo Err_Tool_Make_Part_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fProduct Tool Relationship"
    stLinkCriteria = "[Cat #]=" & " & Me![Cat #] & ":"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Tool_Make_Part_Click:
    Exit Sub

Err_Tool_Make_Part_Click:
    MsgBox Err.Description
    Resume Exit_Tool_Make_Part_Click

End Sub
Private Sub Solid_Work99_Click()
On Error GoTo Err_Solid_Work99_Click

    Dim stAppName As String

    stAppName = "C:sdw99\sldworks.exe"
    Call Shell(stAppName, 1)

Exit_Solid_Work99_Click:
    Exit Sub

Err_Solid_Work99_Click:
    MsgBox Err.Description
    Resume Exit_Solid_Work99_Click

End Sub
Private Sub Back_to_Main_Menu_Click()
On Error GoTo Err_Back_to_Main_Menu_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "Switchboard"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Back_to_Main_Menu_Click:
    Exit Sub
Err_Back_to_Main_Menu_Click:
    MsgBox Err.Description
    Resume Exit_Back_to_Main_Menu_Click

End Sub
Part2: Interface Programs

Option Compare Database
Option Explicit

Private Sub Command28_Click()
On Error GoTo Err_Command28_Click

Dim stDocName As String
stDocName = "rProduct Assembly Detail"
DoCmd.OpenReport stDocName, acPreview

Exit_Command28_Click:
Exit Sub

Err_Command28_Click:
MsgBox Err.Description
Resume Exit_Command28_Click

End Sub
Private Sub Find_Record_Click()
On Error GoTo Err_Find_Record_Click

Screen.PreviousControl.SetFocus
DoCmd.DoMenu acFormBar, acEditMenu, 10, , acMenuVer70

Exit_Find_Record_Click:
Exit Sub

Err_Find_Record_Click:
MsgBox Err.Description
Resume Exit_Find_Record_Click

End Sub
Private Sub Tool_info_for_products_Click()
On Error GoTo Err_Tool_info_for_products_Click

Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "fProduct Tool Relationship"
stLinkCriteria = "[Cat #]=" & "" & Me![Cat #] & ""
DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Tool_info_for_products_Click:
Exit Sub

Err_Tool_info_for_products_Click:
MsgBox Err.Description
Resume Exit_Tool_info_for_products_Click

End Sub
Appendix 3

Private Sub Search_for_record_Click()
On Error GoTo Err_Search_for_record_Click

    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70

Exit_Search_for_record_Click:
    Exit Sub

Err_Search_for_record_Click:
    MsgBox Err.Description
    Resume Exit_Search_for_record_Click

End Sub

Private Sub Product_mould_flow_results_Cmd_Click()
On Error GoTo Err_Product_mould_flow_results_Cmd_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "IPU Product Mould Flow Results"
    stLinkCriteria = "[Cat #]=" & "" & Me![Cat #] & ""
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Product_mould_flow_results_Cmd_Click:
    Exit Sub

Err_Product_mould_flow_results_Cmd_Click:
    MsgBox Err.Description
    Resume Exit_Product_mould_flow_results_Cmd_Click

End Sub

Private Sub Project_Appraisal_Info_Cmd_Click()
On Error GoTo Err_Project_Appraisal_Info_Cmd_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "ISF Project Appraisals"
    stLinkCriteria = "[cat #]=" & "" & Me![Cat #] & ""
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Project_Appraisal_Info_Cmd_Click:
    Exit Sub

Err_Project_Appraisal_Info_Cmd_Click:
    MsgBox Err.Description
    Resume Exit_Project_Appraisal_Info_Cmd_Click

End Sub

Private Sub Product_Design_Questions_Cmd_Click()
On Error GoTo Err_Product_Design_Questions_Cmd_Click

-155-
Appendix 3

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "$PU Product Design Questions"
stLinkCriteria = "[cat #]=" & "" & Me!Cat #) & ""
DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Product_Design_Questions_Cmd_Click:
Exit Sub

Err_Product_Design_Questions_Cmd_Click:
MsgBox Err.Description
Resume Exit_Product_Design_Questions_Cmd_Click

End Sub
Private Sub Product_Designer_s_Cmd_Click()
On Error GoTo Err_Product_Designer_s_Cmd_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "$PU Product Designers"
stLinkCriteria = "[cat #]=" & "" & Me!Cat #) & ""
DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Product_Designer_s_Cmd_Click:
Exit Sub

Err_Product_Designer_s_Cmd_Click:
MsgBox Err.Description
Resume Exit_Product_Designer_s_Cmd_Click

End Sub
Private Sub Assembly_picture_detail_Click()
On Error GoTo Err_Assembly_picture_detail_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "$Product Assembly Detail Subform"
stLinkCriteria = "[Cat #]=" & "" & Me!Cat #) & ""
DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Assembly_picture_detail_Click:
Exit Sub

Err_Assembly_picture_detail_Click:
MsgBox Err.Description
Resume Exit_Assembly_picture_detail_Click

End Sub
Private Sub Product_Assembly_Picture_Cmd_Click()
On Error GoTo Err_Product_Assembly_Picture_Cmd_Click
Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "f Product assembly picture"
stLinkCriteria = "[cat #]=" & "" & Me!{Cat #} & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Product_Assembly_Picture_Cmd_Click:
Exit Sub

Err_Product_Assembly_Picture_Cmd_Click:
MsgBox Err.Description
Resume Exit/Product_Assembly_Picture_Cmd_Click

End Sub
Private Sub Command67_Click()
On Error GoTo Err_Command67_Click

Screen.PreviousControl.SetFocus
DoCmd.FindNext

Exit_Command67_Click:
Exit Sub

Err_Command67_Click:
MsgBox Err.Description
Resume Exit_Command67_Click

End Sub
Private Sub Go_Back_to_Main_Menu_Click()
On Error GoTo Err_Go_Back_to_Main_Menu_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "Switchboard"
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Go_Back_to_Main_Menu_Click:
Exit Sub

Err_Go_Back_to_Main_Menu_Click:
MsgBox Err.Description
Resume Exit_Go_Back_to_Main_Menu_Click

End Sub
Private Sub Run_Word_Click()
On Error GoTo Err_Run_Word_Click

Dim oApp As Object

Set oApp = CreateObject("Word.Application")
oApp.Visible = True
Exit_Run_Word_Click:
Exit Sub

Err_Run_Word_Click:
    MsgBox Err.Description
    Resume Exit_Run_Word_Click

End Sub
Private Sub Solid_Work99_Click()
On Error GoTo Err_Solid_Work99_Click
    Dim stAppName As String

    stAppName = "C:\sw99\solidworks.exe"
    Call Shell(stAppName, 1)

Exit_Solid_Work99_Click:
Exit Sub

Err_Solid_Work99_Click:
    MsgBox Err.Description
    Resume Exit_Solid_Work99_Click

End Sub
Appendix 3

Part 3: Programs for Tool database management

Option Compare Database
Option Explicit

Private Sub Command22_Click()
On Error GoTo Err_Command22_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fSPU Tool Components"
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Command22_Click:
    Exit Sub

Err_Command22_Click:
    MsgBox Err.Description
    Resume Exit_Command22_Click

End Sub
Private Sub Tool_manufacturing_information_Click()
On Error GoTo Err_Tool_manufacturing_information_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fSPU Tool Manufacturing Information"
    stLinkCriteria = "[Tool Code]=" & "&" & Me!"Tool code" & "&"
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Tool_manufacturing_information_Clic:
    Exit Sub

Err_Tool_manufacturing_information_Clic:
    MsgBox Err.Description
    Resume Exit_Tool_manufacturing_information_Clic

End Sub

Private Sub Form_Open(Cancel As Integer)
DoCmd.Maximize

End Sub

Private Sub Tool_code_LostFocus()
DoCmd.RunCommand acCmdSaveRecord

End Sub

Private Sub Tool_component_information_Click()
On Error GoTo Err_Tool_component_information_Click

-159-
Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "IPU Tool components"
stLinkCriteria = "[Tool Code]=" & "" & Me!Tool code] & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Tool_component_information_Click:
    Exit Sub

Err_Tool_component_information_Click:
    MsgBox Err.Description
    Resume Exit_Tool_component_information_Click

End Sub
Private Sub Command35_Click()
On Error GoTo Err_Command35_Click
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "tTool Components"
stLinkCriteria = "[Tool Code]=" & "" & Me!Tool code] & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command35_Click:
    Exit Sub

Err_Command35_Click:
    MsgBox Err.Description
    Resume Exit_Command35_Click

End Sub
Private Sub Command36_Click()
On Error GoTo Err_Command36_Click
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "tTool Components"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command36_Click:
    Exit Sub

Err_Command36_Click:
    MsgBox Err.Description
    Resume Exit_Command36_Click

End Sub
Private Sub Command37_Click()
On Error GoTo Err_Command37_Click

    Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "tTool Components1"
stLinkCriteria = "[Component code]=" & Me!tTool code
DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Command37_Click:
Exit Sub

Err_Command37_Click:
MsgBox Err.Description
Resume Exit_Command37_Click

End Sub
Private Sub cmd_manufacturing_Click()
On Error GoTo Err_cmd_manufacturing_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "tPU Manufacturing Information (Tools)"
stLinkCriteria = "[Tool code]=" & "" & Me!tTool code] & ""
DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_cmd_manufacturing_Click:
Exit Sub

Err_cmd_manufacturing_Click:
MsgBox Err.Description
Resume Exit_cmd_manufacturing_Click

End Sub
Private Sub Command49_Click()
On Error GoTo Err_Command49_Click

Screen.PreviousControl.SetFocus
DoCmd.DoMenuitem acFormBar, acEditMenu, 10, , acMenuVer70

Exit_Command49_Click:
Exit Sub

Err_Command49_Click:
MsgBox Err.Description
Resume Exit_Command49_Click

End Sub
Private Sub Products_made_on_tool_Cmd_Click()
On Error GoTo Err_Products_made_on_tool_Cmd_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "tProduct Tool Relationship"
Appendix 3

stLinkCriteria = "[Tool code]=" & "" & Me![Tool code] & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Products_made_on_tool_Cmd_Click:
 Exit Sub

Err_Products_made_on_tool_Cmd_Click:
 MsgBox Err.Description
 Resume Exit_Products_made_on_tool_Cmd_Click

End Sub
Private Sub Moulding_machine_info_Cmd_Click()
 On Error GoTo Err_Moulding_machine_info_Cmd_Click

 Dim stDocName As String
 Dim stLinkCriteria As String

 stDocName = "IPU Tool Factory Production Machines"

 stLinkCriteria = "[Tool code]=" & "" & Me![Tool code] & """
 DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Moulding_machine_info_Cmd_Click:
 Exit Sub

Err_Moulding_machine_info_Cmd_Click:
 MsgBox Err.Description
 Resume Exit_Moulding_machine_info_Cmd_Click

End Sub
Private Sub Tool_Material_Cmd_Click()
 On Error GoTo Err_Tool_Material_Cmd_Click

 Dim stDocName As String
 Dim stLinkCriteria As String

 stDocName = "IPU Tool Materials"

 stLinkCriteria = "[Tool code]=" & "" & Me![Tool code] & ""
 DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Tool_Material_Cmd_Click:
 Exit Sub

Err_Tool_Material_Cmd_Click:
 MsgBox Err.Description
 Resume Exit_Tool_Material_Cmd_Click

End Sub
Private Sub Mould_Base_Info_Cmd_Click()
 On Error GoTo Err_Mould_Base_Info_Cmd_Click

 Dim stDocName As String
 Dim stLinkCriteria As String
Appendix 3

stDocName = "tSF Mould Base Types"

stLinkCriteria = "[Mould base code]=" & Me!"Mould base code"
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Mould_Base_Info_Cmd_Click:
Exit Sub

Err_Mould_Base_Info_Cmd_Click:
MsgBox Err.Description
Resume Exit_Mould_Base_Info_Cmd_Click

End Sub
Private Sub Command59_Click()
On Error GoTo Err_Command59_Click

DoCmd.GoToRecord , , acNext

Exit_Command59_Click:
Exit Sub

Err_Command59_Click:
MsgBox Err.Description
Resume Exit_Command59_Click

End Sub
Private Sub Tool_designers_Info_Cmd_Click()
On Error GoTo Err_Tool_designers_Info_Cmd_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "tSF Tool Designers"

stLinkCriteria = "[Tool code]=" & "" & Me!"Tool code" & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Tool_designers_Info_Cmd_Click:
Exit Sub

Err_Tool_designers_Info_Cmd_Click:
MsgBox Err.Description
Resume Exit_Tool_designers_Info_Cmd_Click

End Sub
Private Sub Command61_Click()
On Error GoTo Err_Command61_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "tSF Tool Designers"

stLinkCriteria = "[Tool code]=" & "" & Me!"Tool code" & ""
DoCmd.OpenForm stDocName, , , stLinkCriteria
Appendix 3

Exit_Command61_Click:
    Exit Sub

Err_Command61_Click:
    MsgBox Err.Description
    Resume Exit_Command61_Click

End Sub
Private Sub Command62_Click()
On Error GoTo Err_Command62_Click
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "f Tool Designers"

    stLinkCriteria = "[Tool code]=" & "" & Me![Tool code] & ""
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Command62_Click:
    Exit Sub

Err_Command62_Click:
    MsgBox Err.Description
    Resume Exit_Command62_Click

End Sub
Private Sub Tool_Designers_Cmd_Click()
On Error GoTo Err_Tool_Designers_Cmd_Click
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fPU Tool Designers"

    stLinkCriteria = "[Tool code]=" & "" & Me![Tool code] & ""
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Tool_Designers_Cmd_Click:
    Exit Sub

Err_Tool_Designers_Cmd_Click:
    MsgBox Err.Description
    Resume Exit_Tool_Designers_Cmd_Click

End Sub
Private Sub Tool_trials_information_Cmd_Click()
On Error GoTo Err_Tool_trials_information_Cmd_Click
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "FSF Tool Work Request and Production Trial Records"

    stLinkCriteria = "[Tool code]=" & "" & Me![Tool code] & ""

-164-
Appendix 3

DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Tool_trials_information_Cmd_Click:
   Exit Sub

Err_Tool_trials_information_Cmd_Click:
   MsgBox Err.Description
   Resume Exit_Tool_trials_information_Cmd_Click

End Sub
Private Sub Command66_Click()
On Error GoTo Err_Command66_Click

   Dim stDocName As String
   Dim stLinkCriteria As String

   stDocName = "Product Tool Relationship"
   stLinkCriteria = "[Cat #]=" & " & Me![Tool code] & ""
   DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command66_Click:
   Exit Sub

Err_Command66_Click:
   MsgBox Err.Description
   Resume Exit_Command66_Click

End Sub
Private Sub Command67_Click()
On Error GoTo Err_Command67_Click

   Dim stDocName As String
   Dim stLinkCriteria As String

   stDocName = "Product Tool Relationship"
   stLinkCriteria = "[Tool code]=" & " & Me![Tool code] & ""
   DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command67_Click:
   Exit Sub

Err_Command67_Click:
   MsgBox Err.Description
   Resume Exit_Command67_Click

End Sub
Private Sub Back_to_Main_Menu_Click()
On Error GoTo Err_Back_to_Main_Menu_Click

   Dim stDocName As String
   Dim stLinkCriteria As String

   stDocName = "Switchboard"
   DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Back_to_Main_Menu_Click:
    Exit Sub

Err_Back_to_Main_Menu_Click:
    MsgBox Err.Description
    Resume Exit_Back_to_Main_Menu_Click

End Sub
Private Sub Solid_Work99_Click()
    On Error GoTo Err_Solid_Work99_Click
    Dim stAppName As String

    stAppName = "C:\sw99\sldworks.exe"
    Call Shell(stAppName, 1)
Exit_Solid_Work99_Click:
    Exit Sub

Err_Solid_Work99_Click:
    MsgBox Err.Description
    Resume Exit_Solid_Work99_Click

End Sub
Private Sub Command71_Click()
    On Error GoTo Err_Command71_Click
    Call Shell("NOTEPAD.EXE", 1)
Exit_Command71_Click:
    Exit Sub

Err_Command71_Click:
    MsgBox Err.Description
    Resume Exit_Command71_Click

End Sub
Appendix 3

Part 4: Programs for manufacturing information management

Option Compare Database
Option Explicit

Private Sub Command0_Click()
On Error GoTo Err_Command0_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fWorkshop Machinery"
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Command0_Click:
    Exit Sub

Err_Command0_Click:
    MsgBox Err.Description
    Resume Exit_Command0_Click
End Sub

Private Sub Command1_Click()
On Error GoTo Err_Command1_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "FSF Tool Work Request and Production Trial Records"
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Command1_Click:
    Exit Sub

Err_Command1_Click:
    MsgBox Err.Description
    Resume Exit_Command1_Click
End Sub

Private Sub Command2_Click()
On Error GoTo Err_Command2_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "f Workshop (tool trials)"
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Command2_Click:
    Exit Sub

Err_Command2_Click:
    MsgBox Err.Description
    Resume Exit_Command2_Click
End Sub
Private Sub Sodic_wire_cut_info_Cmd_Click()
On Error GoTo Err_Sodic_wire_cut_info_Cmd_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fSF Sodic Wire Cut"
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit_Sodic_wire_cut_info_Cmd_Click:
    Exit Sub

Err_Sodic_wire_cut_info_Cmd_Click:
    MsgBox Err.Description
    Resume Exit_Sodic_wire_cut_info_Cmd_Click

End Sub
Private Sub Factory_Production_Machinery_Cmd_Click()
On Error GoTo Err.Factory_Production_Machinery_Cmd_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fFactory Machinery"
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit.Factory_Production_Machinery_Cmd_Click:
    Exit Sub

Err.Factory_Production_Machinery_Cmd_Click:
    MsgBox Err.Description
    Resume Exit.Factory_Production_Machinery_Cmd_Click

End Sub
Private Sub Machine.Tools_Info_Cmd_Click()
On Error GoTo Err.Machine.Tools_Info_Cmd_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "fSF Machine Tools"
    DoCmd.OpenForm stDocName, , stLinkCriteria

Exit.Machine.Tools_Info_Cmd_Click:
    Exit Sub

Err.Machine.Tools_Info_Cmd_Click:
    MsgBox Err.Description
    Resume Exit.Machine.Tools_Info_Cmd_Click

End Sub
Appendix 4: The Integrated CAD/CAPP/CAM System Program

Source Code for Process Planning for sheet metal cutting

```c
// add assistant path */
#include "lc.h"
#include "user1.h"
#include "user2.h"

void inner_path_2(struct RINGS *info)
{
    struct ELEMENTS *p1,*p2,*p0,*top;
    float xs,ys,xe,ye,ea,sa,r,xc,yc,ra,da,dx,dy;
    float k,k0,xm,ym,x1,y1,x2,y2,sa1,ea2;
    float len_i,len_o,ALFA1,ALFA2;
    int kk,clock,shape,range;

    clock=info->CLOCK;
    k=info->members;
    dx=info->x_max-info->x_min;
    dy=info->y_max-info->y_min;
    ALFA1=120;ALFA2=60;
    len_i=3;len_o=3;
    if((dx<2*len_i)&&(dy<2*len_i)) len_i=len_i/2;
    p1=info->p_elem;
    if((p1->add==1)&&(p1->type!=CIRCLE))
    {
        out_path_1(info);
        return;
    }
    p0=(struct ELEMENTS *)malloc(LEN1);
    if(p0)
    {
        MessageBox(hWnd,"Memory Error!",MB_ICONSTOP,MB_OK);
        exit(0);
    }
    info->p_elem=p0;
    p0->prior_e=NULL;
    if(p1->type==CIRCLE)
    {
        xc=p1->graph.circle.x_center;
        yc=p1->graph.circle.y_center;
        ra=p1->graph.circle.radius;
        info->p_elem=p0;
        if(ra>4)
        {
            // if(ra>10) r=len_i;
            if(r>10)
            {
                r=16;
            }
            else r=ra/2;
            if(p1->add==2)
            {
                xm=xc+ra-r;
                sa=360-ALFA1;
                ea=360;
                xe=xc+ra;
            }
        }
```

else{
    xm=xc-ra+r;
    sa=180-ALFA1;
    ea=180;
    xe=xc-ra;
    ym=yc;
    x1=xm+r*cos(sa*PI/180);
    y1=ym+r*sin(sa*PI/180);
    p0->num=0;
    p0->add=0;     // half circular
    p0->type=ARC;
    p0->graph.arc.aradius=r;
    p0->graph.arc.x_acenter=xm;//xc-ra+r;
    p0->graph.arc.y_acenter=ym;
    p0->graph.arc.x_end=xe;//xc-ra;
    p0->graph.arc.y_end=yc;
    p0->graph.arc.angle=ea;//180;
    p0->graph.arc.size=sa;//180-ALFA1;
    p0->graph.arc.dangle=ALFA1;
    p0->graph.arc.shape=3;
    p0->graph.arc.x_start=x1;
    p0->graph.arc.y_start=y1;
    p0->next_e=p1;
    p1->prior_e=p0;
    kk++;
    p0=(struct ELEMENTS *)malloc(LEN1);
    if(p0)
    
    MessageBox(hWnd,"memory error!","",MB_ICONSTOP|MB_OK);
    exit(0);
    }
    p0->num=kk;
    p0->add=0;
    p0->type=ARC;
    p0->graph.arc.aradius=r;
    if(p1->add==2)
    
    sa=0;
    ea=ALFA2;
    }
else{
    sa=180;
    ea=180+ALFA2;
    }
    x2=xm+r*cos(ea*PI/180);  // other half circular
    y2=ym+r*sin(ea*PI/180);
    p0->graph.arc.x_acenter=xm;
    p0->graph.arc.y_acenter=ym;
    p0->graph.arc.x_start=xe;
    p0->graph.arc.y_start=yc;
    p0->graph.arc.eangle=ea;
    p0->graph.arc.size=sa;
    p0->graph.arc.dangle=ALFA2;
    p0->graph.arc.shape=3;
p0->graph.arc.x_end=x2;
p0->graph.arc.y_end=y2;
p1->next_e=p0;
p0->prior_e=p1;
p0->next_e=NULL;
kk++;
} else {
p0->num=0;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_start=xc;
p0->graph.line.y_start=yc;
if(p1->add==2) p0->graph.line.x_end=xc+ra;
else p0->graph.line.x_end=xc-ra;
p0->graph.line.y_end=yc;
p0->next_e=p1;
p1->prior_e=p0;
k++;
struct ELEMENTS *s)malloc(LEN1);
if(!p0) {
    MessageBox(hWnd, "memory error!!","",MB_ICONSTOP|MB_OK);
    exit(0);
}
p0->num=kk;
p0->add=0;
p0->type=LINE;
if(p1->add==2) p0->graph.line.x_start=xc+ra;
else p0->graph.line.x_start=xc-ra;
p0->graph.line.y_start=yc;
p0->graph.line.x_end=xc;
p0->graph.line.y_end=yc;
p1->next_e=p0;
p0->prior_e=p1;
p0->next_e=NULL;
k++;
}
info->members=kk;
return;
}
if(p1->type==LINE) /* start with line */
{
x=x1->graph.line.x_start;
y=y1->graph.line.y_start;
x=x1->graph.line.x_end;
y=y1->graph.line.y_end;
xm=0.5*(xs+xe);
yn=0.5*(ys+ye);
// r=len_i;
r=16.;
k=slope_angle(xs,ys,xe,ye);
if(clock==1)
{
    xc=xm-r*sin(k);
    yc=ym+r*cos(k);
k0=slope_angle(xc,yc,xm,ym);
sa1=k0*ALFA1*PI/180;
if(sa1<0) sa1=sa1+2*PI;
ea2=k0+ALFA2*PI/180;
if(ea2>2*PI) ea2=ea2-2*PI;
}
if(clock==0)
{
x=xc+r*cos(sa1);
y=yc+r*sin(sa1);
x1=x1+r*cos(sa1);
y1=y1+r*sin(sa1);
x2=x2+r*cos(ea2);
y2=y2+r*sin(ea2);
p0->num=0;
p0->type=ARC;
p0->graph.arc.radius=r;
p0->graph.arc.x_center=x;
p0->graph.arc.y_center=y;
p0->graph.arc.x_start=xm;
p0->graph.arc.y_start=ym;
p0->graph.arc.x_end=xm;
p0->graph.arc.y_end=ym;
p0->graph.arc.angle=k0*180/PI;
p0->graph.arc.angle=sa1*180/PI;
p0->graph.arc.angle=ALFA1;
if(clock==0) p0->graph.arc.shape=2;
else p0->graph.arc.shape=3;
p0->graph.arc.x_start=x1;
p0->graph.arc.y_start=y1;
end=p0;
// new_store(p0);
k++;
p2=(struct ELEMENTS *)malloc(LEN1);
if(!p2)
{
    MessageBox(hWnd, "memory error!", "", MB_ICONSTOP|MB_OK);
    exit(0);
}
p2->add=1;
p2->num=1;
p2->type=LINE;
p2->graph.line.x_start=xm;
p2->graph.line.y_start=ym;
p1->graph.line.x_start=xm;
p1->graph.line.y_start=ym;
p2->graph.line.x_end=x;
p2->graph.line.y_end=y;
break_LA(p1,p2,p0);
k++;
p0=(struct ELEMENTS *)malloc(LEN1);
if(!p0)
MessageBox(hWnd,"memory error!",",",MB_ICONSTOP,MB_OK);
exit(0);
}
p0->num=kk+2;
p0->add=0;
p0->type=ARC;
p0->graph.arc.radius=r;
p0->graph.arc.x_center=x;
p0->graph.arc.y_center=y;
p0->graph.arc.x_start=x;
p0->graph.arc.y_start=y;
p0->graph.arc.x_end=x2;
p0->graph.arc.y_end=y2;
p0->graph.arc.angle=ea2*180/PI;
p0->graph.arc.angle=k0*180/PI;
p0->graph.arc.dangle=ALFA2;
if(clock==0) p0->graph.arc.shape=2;
else p0->graph.arc.shape=3;
p1->next_e=p0;
p0->prior_e=p1;
p0->next_e=NULL;
// new_store(p0);
k++;
info->members=kk;
}
if(p1->type==ARC)
{
x=p1->graph.arc.x_start;
y=p1->graph.arc.y_start;
x1=p1->graph.arc.x_end;
y1=p1->graph.arc.y_end;
x2=p1->graph.arc.x_center;
y2=p1->graph.arc.y_center;
x3=p1->graph.arc.radius;
y3=p1->graph.arc.angle;
shape=p1->graph.arc.shape;
if(shape==3){
xm=x+c*cos((sa+da2)*PI/180);
ym=y+c*cos((sa+da2)*PI/180);
mangle=sa+da2;
}
if(shape==2){
xm=x+c*cos((sa-da2)*PI/180);
ym=y+c*cos((sa-da2)*PI/180);
mangle=sa-da2;
}
k0=slope_angle(xc,yc,xm,ym);
if(clock==1){
k=k0+PI/2;
if(k>2*PI) k=k-2*PI;
}
if(clock==0){
k=k0-PI/2;
}
if(k0<0) k=k+2*PI;
}
if((clock==0)&&(shape==2))||((clock==1)&&(shape==3))
{
  // t=len_i;
  r=16;
  if(clock==1)
  {
    xc=xm+r*sin(k);
    yc=ym+r*cos(k);
    sa1=k0-ALFA1*PI/180;
    if(sa1<0) sa1=sa1+2*PI;
    ca2=k0-ALFA2*PI/180;
    if(ca2>2*PI) ca2=ca2-2*PI;
  }
  if(clock==0)
  {
    xc=xm+r*cos(sa1);
    yc=ym+r*sin(sa1);
    x2=xc+r*cos(ea2);
    y2=yc+r*sin(ea2);
    p0->num=0;
    p0->add=0;
    p0->type=ARC;
    p0->graph.arc.aradius=r;
    p0->graph.arc.x_acenter=xc;
    p0->graph.arc.y_acenter=yc;
    p0->graph.arc.x_end=xm;
    p0->graph.arc.y_end=ym;
    p0->graph.arc.angle=k0*180/PI;
    p0->graph.arc.sangle=sa1*180/PI;
    p0->graph.arc.dangle=ALFA1;
    if(shape==3) p0->graph.arc.shape=3;
    else p0->graph.arc.shape=2;
    p0->graph.arc.x_start=x1;
    p0->graph.arc.y_start=y1;
    // new_store(p0);
    kk++;
    p2=(struct ELEMENTS *)malloc(LEN1);
    if(!p2)
      
      MessageBox(hWnd, "memory error!", ",", MB_ICONSTOP|MB_OK);
    exit(0);
  }      
  p2->add=1;
  p2->num=1;
  p2->type=ARC;
  p2->graph.arc.x_start=xm;
  p2->graph.arc.y_start=ym;
p2->graph.arc.x_end=xe;
p2->graph.arc.y_end=ye;
p2->graph.arc.angle=mangle;
p2->graph.arc.eangle=ea;
p2->graph.arc.dangle=da/2;
p2->graph.arc.aradius=ra;
p2->graph.arc.x_acentor=p1->graph.arc.x_acentor;
p2->graph.arc.y_acentor=p1->graph.arc.y_acentor;
p2->graph.arc.shape=p1->graph.arc.shape;
p1->graph.arc.x_end=xm;
p1->graph.arc.y_end=ym;
p1->graph.arc.eangle=mangle;
p1->graph.arc.dangle=da/2;
break_LA(p1,p2,p0);
// A_ms_L(p1,p2,start);
 kk++;
p0=((struct ELEMENTS *)malloc(LEN1));
if(!p0)
  [MessageBox(hWnd, "memory error!", "MB_ICONSTOP!MB_OK");
  exit(0);
  ]
p0->num=kk;
p0->add=0;
p0->type=ARC;
p0->graph.arc.aradius=r;
p0->graph.arc.x_acentor=xc;
p0->graph.arc.y_acentor=yc;
p0->graph.arc.x_start=xm;
p0->graph.arc.y_start=ym;
p0->graph.arc.x_end=x2;
p0->graph.arc.y_end=y2;
p0->graph.arc.eangle=ea2*180/PI;
p0->graph.arc.sangle=k0*180/PI;
p0->graph.arc.dangle=ALFA2;
if(shape==3) p0->graph.arc.shape=3;
else p0->graph.arc.shape=2;
p1->next_e=p0;
p0->prior_e=p1;
p0->next_e=NULL;
// new_store(p0);
 kk++;
]
if((clock==0)&&(shape==3)||(clock==1)&&(shape==2))
{ len_o=ra/2;
s1=k;
if(clock==1) ea2=s1-PI;
else ea2=s1+PI;
if(ea2>2*PI) ea2=ea2-2*PI;
if(ea2<0) ea2=ea2+2*PI;
x2=xm+len_o*cos(ea2);
y2=ym+len_o*sin(ea2);
x1=xm+len_o*cos(s1);
y1=ym+len_o*sin(s1);
p0->num=0;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_start=x1;
p0->graph.line.y_start=y1;
p0->graph.line.x_end=xm;
p0->graph.line.y_end=ym;
// new_store(p0);
kk++;
p2=(struct ELEMENTS *)malloc(LEN1);
if(!p2)
{
    MessageBox(hWnd, "memory error!", ",MB_ICONSTOP,MB_OK);
    exit(0);
}
p2->add=1;
p2->num=1;
p2->type=ARC;
p2->graph.arc.arcRadius=ra;
p2->graph.arc.x_acenter=p1->graph.arc.x_acenter;
p2->graph.arc.y_acenter=p1->graph.arc.y_acenter;
p2->graph.arc.x_start=xm;
p2->graph.arc.y_start=ym;
p2->graph.arc.mangle=mangle;
p2->graph.arc.eangle=ea;
p2->graph.arc.dangle=da/2;
p2->graph.arc.shape=p1->graph.arc.shape;
p2->graph.arc.x_end=xe;
p2->graph.arc.y_end=ye;
p1->graph.arc.x_end=xm;
p1->graph.arc.y_end=ym;
p1->graph.arc.eangle=mangle;
p1->graph.arc.dangle=da/2;
break_LA(p1,p2,p0);
// A_ins_L(p1,p2,op);
kk++;
p0=(struct ELEMENTS *)malloc(LEN1);
if(!p0)
{
    MessageBox(hWnd, "memory error!", ",MB_ICONSTOP,MB_OK);
    exit(0);
}
p0->num=kk+2;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_start=xm;
p0->graph.line.y_start=ym;
p0->graph.line.x_end=x2;
p0->graph.line.y_end=y2;
p1->next_e=p0;
p0->prior_e=p1;
p0->next_e=NULL;
// new_store(p0);
kk++; 
}
info->members=kk;
Appendix 4

```c
void out_path_2(struct RINGS *info)
{
  struct ELEMENTS *p1, *p2, *p0, *p3, *top;
  float xs, ys, xe, ye, ra, xc, yc, r, sa1, ea2, da, sa, ea;
  float k0, x1, y1, x2, y2, xm, ym, mangle;
  float ALFA1, ALFA2, len_o=12, len_i=12;
  int kk, clock, shape;
  p1=info->p_elm;
  kk=info->members;
  clock=info->CLOCK;
  ALFA1=120; ALFA2=60;
  if((info->include==1) || (info->include==0))
    {
      if(solid==1) { len_o=15; len_i=15; }
      else { len_o=10; len_i=10; }
    }
  p0=(struct ELEMENTS *)malloc(LEN1);
  if(!p0)
    {
      MessageBox(hWnd, "memory error!", ",", MB_ICONSTOP|MB_OK);
      exit(0);
    }
  info->p_elm=p0;
  p0->prior_e=NULL;
  if(p1->type==CIRCLE)
    {
      xc=p1->graph.circle.x_center;
      yc=p1->graph.circle.y_center;
      ra=p1->graph.circle.radius;
      if(ra<10) r=len_o;
      else r=0.5+ra;
      if(r>10) r=10;
      p0->num=0;
      p0->add=0;
      p0->type=LINE;
      p0->graph.line.x_end=xc+ra;
      p0->graph.line.y_end=yc;
      p0->graph.line.x_start=xc+ra;
      p0->graph.line.y_start=yc-r;
      kk++;
      p0->next_e=p1;
      p1->prior_e=p0;
      p0=(struct ELEMENTS *)malloc(LEN1);
      if(!p0)
        {
          MessageBox(hWnd, "memory error!", ",", MB_ICONSTOP|MB_OK);
          exit(0);
        }
      p0->num=kk;
      p0->add=0;
      p0->type=LINE;
      p0->graph.line.x_end=xc+ra;
      p0->graph.line.y_end=yc+r;
      p0->graph.line.x_start=xc+ra;
      p0->graph.line.y_start=yc;
    }
```
p1->next_e=p0;
p0->prior_e=p1;
p0->next_e=NULL;
kk++;
info->members=kk;
}
if(p1->type==LINE)
{
xz=p1->graph.line.x_end;
yz=p1->graph.line.y_end;
xs=p1->graph.line.x_start;
ys=p1->graph.line.y_start;
xm=0.5*(xz+xz);
ym=0.5*(yz+yz);
// r=len_i;

r=16.;
k=slope_angle(xs,ys,xz,yz);
if(clock==0)
{
    xc=xm+r*sin(k);
yc=ym+r*cos(k);
k0=slope_angle(xc,yc,xm,ym);
sal=k0-ALFA1*PI/180;
if(sal<0) sal=sal+2*PI;
    ea2=k0+ALFA2*PI/180;
if(ea2>2*PI) ea2=ea2-2*PI;
}
if(clock==1)
{
    xc=xm+r*sin(k);
yc=ym+r*cos(k);
k0=slope_angle(xc,yc,xm,ym);
sal=k0+ALFA1*PI/180;
if(sal>2*PI) sal=sal-2*PI;
    ea2=k0-ALFA2*PI/180;
if(ea2<0) ea2=ea2+2*PI;
}
x1=xc+r*cos(sal);
y1=yc+r*sin(sal);
x2=xc+r*cos(ea2);
y2=yc+r*sin(ea2);
p0->num=0;
p0->add=0;
p0->type=ARC;
p0->graph.arc.arc.radius=r;
p0->graph.arc.arc.center=xc;
p0->graph.arc.arc.y_center=yc;
p0->graph.arc.arc.x_end=xm;
p0->graph.arc.arc.y_end=ym;
p0->graph.arc.arc.angle=k0*180/PI;
p0->graph.arc.arc_sangle=sal*180/PI;
p0->graph.arc.arc_aangle=ALFA1;
if(clock==1) p0->graph.arc.shape=2;
else p0->graph.arc.shape=3;
p0->graph.arc.arc.x_start=x1;
p0->graph.arc.arc.y_start=y1;
struct ELEMENTS *malloc(len1);
if(p2)
{
    MessageBox(hWnd, "memory error!", "MB_ICONSTOP|MB_OK");
    exit(0);
}
p2->add=1;
p2->num=1;
p2->type=LINE;
p2->graph.line.x_start=xm;
p2->graph.line.y_start=ym;
p1->graph.line.x_end=xm;
p1->graph.line.y_end=ym;
p2->graph.line.x_end=xe;
p2->graph.line.y_end=ye;
break LA(p1,p2,p0);
kk++;
p0=(struct ELEMENTS *)malloc(len1);
if(!p0)
{
    MessageBox(hWnd, "memory error!", "MB_ICONSTOP|MB_OK");
    exit(0);
}
p0->num=kk+2;
p0->add=0;
p0->type=ARC;
p0->graph.arc.radius=r;
p0->graph.arc.x_center=xcc;
p0->graph.arc.y_center=yc;
p0->graph.arc.x_start=xm;
p0->graph.arc.y_start=ym;
p0->graph.arc.x_end=x2;
p0->graph.arc.y_end=y2;
p0->graph.arc.angle=ea2*180/PI;
p0->graph.arc.sangle=k0*180/PI;
p0->graph.arc.dangle=ALFA2;
if(clock==1) p0->graph.arc.shape=2;
else p0->graph.arc.shape=3;
p1->next_e=p0;
p0->prior_e=p1;
p0->next_e=NULL;
kk++;
info->members=kk;
}
if(p1->type==ARC)
{
    xs=p1->graph.arc.x_start;
    ys=p1->graph.arc.y_start;
    xe=p1->graph.arc.x_end;
    ye=p1->graph.arc.y_end;
    xc=p1->graph.arc.x_center;
    yc=p1->graph.arc.y_center;
    ra=p1->graph.arc.radius;
    sa=p1->graph.arc.sangle;
    ea=p1->graph.arc.eangle;
Appendix 4

da=p1->graph.arc.dangle;
shape=p1->graph.arc.shape;
if(shape==3){
xm=xc+ra*cos((sa+da)/2)*PI/180);
ym=yc+ra*sin((sa+da)/2)*PI/180);
angle=sa+da/2;
}
if(shape==2){
xm=xc+ra*cos((sa-da)/2)*PI/180);
ym=yc+ra*sin((sa-da)/2)*PI/180);
angle=sa-da/2;
}
k=slope_angle(xc,yc,xm,ym);
if(clock==1){
k=k+PI/2;
if(k>2*PI) k=k-2*PI;
}
if(clock==0){
k=k-PI/2;
if(k<0) k=k+2*PI;
}
if((clock==0)&&(shape==2))if((clock==1)&&(shape==3))
{len_o=ra;
  //
if(len_o>10) len_o=10;
ea2=k;
if(clock==1) sa1=ea2-PI;
else sa1=ea2+PI;
if(sa1>2*PI) sa1=sa1-2*PI;
if(sa1<0) sa1=sa1+2*PI;
xb=xm+len_o*cos(ea2);
yb=ym+len_o*sin(ea2);
x1=xm+len_o*cos(sa1);
y1=ym+len_o*sin(sa1);
p0->num=0;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_start=x1;
p0->graph.line.y_start=y1;
p0->graph.line.x_end=xm;
p0->graph.line.y_end=ym;
  //
new_store(p0);
kk++;}
p2=(struct ELEMENTS *)malloc(LEN1);
if(p2)
{
  MessageBox(hWnd, "memory error!", ":", MB_ICONSTOP|MB_OK);
exi(0);
}
p2->add=1;
p2->num=1;
p2->type=ARC;
p2->graph.arc.radius=ra;
p2->graph.arc.x_center=p1->graph.arc.x_center;
p2->graph.arc.y_center=p1->graph.arc.y_center;
p2->graph.arc.x_start=xm;
p2->graph.arc.y_start=ym;
p2->graph.arc.sangle=mangle;
p2->graph.arc.eangle=ea;
p2->graph.arc.dangle=da/2;
p2->graph.arc.shape=p1->graph.arc.shape;
p2->graph.arc.x_end=xe;
p2->graph.arc.y_end=ye;
p1->graph.arc.x_end=xm;
p1->graph.arc.y_end=ym;
p1->graph.arc.eangle=mangle;
p1->graph.arc.dangle=da/2;
// A_ins_L(p1,p2,top);
break_LA(p1,p2,p0);
kk++;
p0=(struct ELEMENTS *)malloc(LEN1);
if(p0)
{
    MessageBox(hWnd, "memory error!", "MB_ICONSTOP|MB_OK");
    exit(0);
}
p0->num=kk+2;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_start=xm;
p0->graph.line.y_start=ym;
p0->graph.line.x_end=x2;
p0->graph.line.y_end=y2;
p1->next_e=p0;
p0->prior_e=p1;
p0->next_e=NULL;
// new_store(p0);
kk++;
}
i=((clock==0)&&(shape==3))&&(clock==1)&&(shape==2))
{
    // r=0.5*r;
    // if(r<len_i) r=len_i;
    r=16.;
    if(clock==0)
    {
        xc=xm+r*sin(k);
        yc=ym-r*cos(k);
        sa1=k0-ALFA1*PI/180;
        if(sa1<0) sa1=sa1+2*PI;
        ea2=k0+ALFA2*PI/180;
        if(ea2>2*PI) ea2=ea2-2*PI;
    }
    if(clock==1)
    {
        xc=xm-r*sin(k);
        yc=ym+r*cos(k);
        sa1=k0+ALFA1*PI/180;
        if(sa1>2*PI) sa1=sa1-2*PI;
        ea2=k0-ALFA2*PI/180;
        if(ea2<0) ea2=ea2+2*PI;
    }
x1=xc+r*cos(sal1);
y1=yc+r*sin(sal1);
x2=xc+r*cos(ea2);
y2=yc+r*sin(ea2);
p0->num=0;
p0->add=0;
p0->type=ARC;
p0->graph.arc.radius=r;
p0->graph.arc.x_center=xc;
p0->graph.arc.y_center=yc;
p0->graph.arc.x_end=xm;
p0->graph.arc.y_end=ym;
p0->graph.arc.eangle=k0*180/PI;
p0->graph.arc.sangle=sal1*180/PI;
p0->graph.arc.dangle=ALFA1;
if(shape==3) p0->graph.arc.shape=3;
else p0->graph.arc.shape=2;
p0->graph.arc.x_start=x1;
p0->graph.arc.y_start=y1;
#else new_store(p0);
  kk++;
  p2=(struct ELEMENTS *)malloc(LEN1);
  if(p2)
    MessageBox(hWnd, "memory error!", ",MB_ICONSTOP,MB_OK);
    exit(0);
  }
p2->add=1;
p2->num=1;
p2->type=ARC;
p2->graph.arc.x_start=xm;
p2->graph.arc.y_start=ym;
p2->graph.arc.x_end=xe;
p2->graph.arc.y_end=ye;
p2->graph.arc.sangle=mangle;
p2->graph.arc.eangle=ea;
p2->graph.arc.dangle=da/2;
p2->graph.arc.aradius=ra;
p2->graph.arc.x_center=p1->graph.arc.x_center;
p2->graph.arc.y_center=p1->graph.arc.y_center;
p2->graph.arc.shape=p1->graph.arc.shape;
p1->graph.arc.x_end=xm;
p1->graph.arc.y_end=ym;
p1->graph.arc.eangle=mangle;
p1->graph.arc.dangle=da/2;
break_L_A(p1,p2,p0);
// A_ins_L(p1,p2,top):
nk++;
  p0=(struct ELEMENTS *)malloc(LEN1);
  if(p0)
    MessageBox(hWnd, "memory error!", ",MB_ICONSTOP,MB_OK);
    exit(0);
  }
p0->num=kk;
p0->add=0;
p0->type = ARC;
p0->graph.arc.arc_radius = r;
p0->graph.arc.x_center = xc;
p0->graph.arc.y_center = yc;
p0->graph.arc.x_start = xm;
p0->graph.arc.y_start = ym;
p0->graph.arc.x_end = x2;
p0->graph.arc.y_end = y2;
p0->graph.arc.cangle = ea * 180 / PI;
p0->graph.arc.sangle = ka * 180 / PI;
p0->graph.arc.dangle = ALFA2;
if(shape == 3) p0->graph.arc.shape = 3;
else p0->graph.arc.shape = 2;
p1->next_e = p0;
p0->prior_e = p1;
p0->next_e = NULL;

// new_store(p0);
kk++;
}
info->members = kk;
]
}

void out_path_1(struct RINGS *info)
{
    struct ELEMENTS *p1, *p2, *p0;
    float xs, ys, xe, ye, ra, xc, yc, r, sa1, ea2, da, sa, ea;
    float k1, k2, x1, y1, x2, y2, xm, ym, mangle;
    float ALFA1, ALFA2, len_o = 16, len_i = 16;
    int kk, clock, shape;
    // top = info->p_elem;
    kk = info->members;
    clock = info->CLOCK;
    ALFA1 = 120; ALFA2 = 60;
    if((info->include == -1) && (info->include == 0))
    {
        if(solid == 1) { len_o = 16; len_i = 16; }
    }
    p1 = info->p_elem;
    while(p1)
    {
        p2 = p1;
        p1 = p1->next_e;
    }
    p1 = info->p_elem;
    // last_path = NULL;
    if(p1->type == CIRCLE)
    {
        xc = p1->graph.circle.x_center;
        yc = p1->graph.circle.y_center;
        ra = p1->graph.circle.radius;
        if(ra < 10) r = len_o;
        else r = 0.5 * ra;
        // if(r > 10) r = 10;
        p0 = (struct ELEMENTS *) malloc(LEN1);
        if(!p0)
        {
            MessageBox(hWnd, "memory error!", ",", MB_ICONSTOP | MB_OK);
        }
    }
exit(0);
}
info->p_elem=p0;
// last_path=NULL;
p0->num=0;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_end=xc-ra;
p0->graph.line.y_end=yc;
p0->graph.line.x_start=xc-ra;
p0->graph.line.y_start=yc+r;
// new_store(p0);
p0->prior_e=NULL;
p0->next_e=p1;
p1->prior_e=p0;
kk++;
// C_ins_ring(p1);
p0=(struct ELEMENTS *)malloc(LEN1);
if(!p0)
{
    MessageBox(hWnd, "memory error!", "", MB_ICONSTOP, MB_OK);
    exit(0);
}
p0->num=kk;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_end=xc-ra;
p0->graph.line.y_end=yc-r;
p0->graph.line.x_start=xc-ra;
p0->graph.line.y_start=yc;
// new_store(p0);
p2->next_e=p0;
p0->prior_e=p2;
p0->next_e=NULL;
kk++;
info->members=kk;
return;
}
if(p1->type==LINE)
{
    xe=p1->graph.line.x_end;
    ye=p1->graph.line.y_end;
    xs=p1->graph.line.x_start;
    ys=p1->graph.line.y_start;
    k1=slope_angle(xe, ye, xs, ys);
}
if(p1->type==ARC)
{
    xs=p1->graph.arc.x_start;
    ys=p1->graph.arc.y_start;
    xc=p1->graph.arc.x_center;
    yc=p1->graph.arc.y_center;
    shape=p1->graph.arc.shape;
    k1=slope_angle(xs, ys, xc, yc);
    if(shape==2) k1=k1-PI/2;
    else k1=k1+PI/2;
}
x1=x+\text{len}_o*\cos(k1);
y1=y+\text{len}_o*\sin(k1);
p0=(\text{struct ELEMENTS } *)\text{malloc(LEN1)};
if(p0)
|
MessageBox(hWnd, "memory error!", MB_ICONSTOP|MB_OK);
exit(0);
|
info->p_elem=p0;
//last_path=NULL;
p0->num=1;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_end=x1;
p0->graph.line.y_end=y1;
p0->graph.line.x_start=x1;
p0->graph.line.y_start=y1;
p0->prior_e=NULL;
p0->next_e=p1;
p1->prior_e=p0;
// new_store(p0);
kk++;
// L_ins_ring(p1,top);
if(p2->type==LINE)
|
xe=p2->graph.line.x_end;
ye=p2->graph.line.y_end;
x1=p2->graph.line.x_start;
y1=p2->graph.line.y_start;
k2=slope_angle(x1,y1,xe,ye);
|
if(p2->type==ARC)
|
x1=p2->graph.arc.x_end;
y1=p2->graph.arc.y_end;
x1=p2->graph.arc.x_center;
y1=p2->graph.arc.y_center;
shape=p2->graph.arc.shape;
k2=slope_angle(xe,xe,xc,yc);
if(shape==2) k2=k2+PI/2;
else k2=k2-PI/2;
|
x2=x+\text{len}_o*\cos(k2);
y2=y+\text{len}_o*\sin(k2);
p0=(\text{struct ELEMENTS } *)\text{malloc(LEN1)};
if(p0)
|
MessageBox(hWnd, "memory error!", MB_ICONSTOP|MB_OK);
exit(0);
|
p0->num=kk+1;
p0->add=0;
p0->type=LINE;
p0->graph.line.x_end=x2;
p0->graph.line.y_end=y2;
p0->graph.line.x_start=x0;
p0->graph.line.x_start=ye;
p2->next_e=p0;
p0->prior_e=p2;
p0->next_e=NULL;
// new_store(p3);
kk++; info->members=kk;
return;
}
void break_LA(struct ELEMENTS *p1,struct ELEMENTS *p2,struct ELEMENTS *p0)
/**/
{
struct ELEMENTS *p,*pend;
int j;
p2->next_e=p1->next_e;
p1->next_e->prior_e=p2;
p2->prior_e=p0;
p1->next_e=NULL;
p0->next_e=p2;
j=0;
p=p2;
while(p){
    p->num=++j;
    pend=p;
    p=p->next_e;
}
p1->prior_e=pend;
pend->next_e=p1;
p1->num=++j;
}
/*struct ELEMENTS *add_path(struct ELEMENTS *p1,struct RINGS *info)
{
struct ELEMENTS *p2,*p0,*p3;
float x11,y11,x12,y12,x22,y22;
float x1c1,x1c2,yc2,k,x,y;
float LL=10.0,CL=2.0;
int num1,shape1,shape2;

p2=p1->next_e;
if(p2==NULL) p2=info->p_elem;
if((p1->type==LINE)&&(p2->type==LINE))
{
    x11=p1->graph.line.x_start;
y11=p1->graph.line.y_start;
x12=p1->graph.line.x_end;
y12=p1->graph.line.y_end;
x22=p2->graph.line.x_end;
y22=p2->graph.line.y_end;
}
if((p1->type==LINE)&&(p2->type==ARC))
{
    x11=p1->graph.line.x_start;
y11=p1->graph.line.y_start;
x12=p1->graph.line.x_end;

-186-
y12=p1->graph.line.y_end;
x2c=p2->graph.arc.x_center;
y2c=p2->graph.arc.y_center;
shape2=p2->graph.arc.shape;
if(fabs(xc2-x12)<0.001){
  if(y12>y2c){
    x22=x12+LL;
    y22=y12;
  }
  if(y12<y2c){
    x22=x12-LL;
    y22=y12;
  }
}
if(fabs(x12-xc2)>0.001){
  if(y12==y2c){
    if(x12==xc2) {
      if(shape2==3) y22=y12+LL;
      else y22=y12+LL;
    } 
    else y22=y12-LL;
  } 
  if(x12==xc2) {
    if(shape2==3) y22=y12+LL;
    else y22=y12-LL;
  }
  x22=x12;
} else{
  k=(y12-y2c)/(x12-xc2);
  k=tan(fabs(k));
  if(((x12-xc2)>0)&&((y12-y2c)>0)){
    if(shape2==3){
      x22=x12+LL*sin(k);
      y22=y12+LL*cos(k);
    } 
    if(shape2==2){
      x22=x12+LL*sin(k);
      y22=y12-LL*cos(k);
    } 
  } 
  if(((x12-xc2)<0)&&((y12-y2c)<0)){
    if(shape2==3){
      x22=x12-LL*sin(k);
      y22=y12-LL*cos(k);
    } 
    if(shape2==2){
      x22=x12-LL*sin(k);
      y22=y12+LL*cos(k);
    } 
  } 
  if(((x12-xc2)<0)&&((y12-y2c)>0)){
    if(shape2==3){
      x22=x12+LL*sin(k);
      y22=y12-LL*cos(k);
    } 
    if(shape2==2){
      x22=x12-LL*sin(k);
      y22=y12+LL*cos(k);
    } 
  } 
  if(((x12-xc2)<0)&&((y12-y2c)<0)){
    if(shape2==3){
      x22=x12+LL*sin(k);
      y22=y12+LL*cos(k);
    } 
  }
if(shape2==2){
    x22=x12-LL*sin(k);
y22=y12+LL*cos(k);
}
}
if(((x12-xc2)>0)&&(y12-yc2)<0)){
    if(shape2==3){
        x22=x12+LL*sin(k);
y22=y12+LL*cos(k);
    }
    if(shape2==2){
        x22=x12-LL*sin(k);
y22=y12-LL*cos(k);
    }
}
}
}

if((p1->type==ARC)&&(p2->type==ARC))
{
    x12=p1->graph.arc.x_end;
y12=p1->graph.arc.y_end;
    xc1=p1->graph.arc.x_center;
yc1=p1->graph.arc.y_center;
    xc2=p2->graph.arc.x_center;
yc2=p2->graph.arc.y_center;
    shape1=p1->graph.arc.shape;
    shape2=p2->graph.arc.shape;
    if(fabs(xc1-x12)<=0.001){
        if(y12>yc1){
            if(shape1==2) x11=x12-LL;
            if(shape1==3) x11=x12+LL;
y11=y12;
        }
        if(y12<yc1){
            if(shape1==2) x11=x12+LL;
            if(shape1==3) x11=x12-LL;
y11=y12;
        }
    }
    if(fabs(x12-xc1)>0.001){
        if(y12==yc1){
            if(x12>xc1){
                if(shape1==2) y11=y12+LL;
                if(shape1==3) y11=y12-LL;
            }
            if(x12<xc1){
                if(shape1==2) y11=y12-LL;
                if(shape1==3) y11=y12+LL;
            }
        }x11=x12;
    }
    else {
        k=(y12-yc1)/(x12-xc1);
k=atan(fabs(k));
        if(((x12-xc1)>0)&&(y12-yc1)>0)){
if(shape1==3) {
    x11=x12+LL*sin(k);
    y11=y12-LL*cos(k);
    }
if(shape1==2) {
    x11=x12-LL*sin(k);
    y11=y12-LL*cos(k);
    }
if(shape1==3) {
    x11=x12+LL*sin(k);
    y11=y12+LL*cos(k);
    }
}
if(((x12-xc1)<0)&&((y12-yc1)<0)) {
    if(shape1==2) {
        x11=x12+LL*sin(k);
        y11=y12-LL*cos(k);
        }
    if(shape1==3) {
        x11=x12-LL*sin(k);
        y11=y12-LL*cos(k);
        }
    }
if(((x12-xc1)>0)&&((y12-yc1)<0)) {
    if(shape1==2) {
        x11=x12+LL*sin(k);
        y11=y12+LL*cos(k);
        }
    if(shape1==3) {
        x11=x12-LL*sin(k);
        y11=y12-LL*cos(k);
        }
    }
}
if(fabs(xc2-x12)<=0.001) {
    if(y12>yc2) {
        if(shape2==2) x22=x12+LL;
        if(shape2==3) x22=x12-LL;
        y22=y12;
        }
    if(y12<yc2) {
        if(shape2==2) x22=x12-LL;
        if(shape2==3) x22=x12+LL;
        y22=y12;
        }
    }
if(fabs(x12-xc2)>0.001) {
    if(y12==yc2) {
        if(x12>xc2) {
            
        }
    }
if(fabs(xc2-x12)>0.001) {
    if(y12==yc2) {
        if(x12>xc2) {
            
        }
    }
}
if(shape2==2) y22=y12-LL;
if(shape2==3) y22=y12+LL;
}
if(x12<xc2){
  if(shape2==2) y22=y12+LL;
  if(shape2==3) y22=y12-LL;
}
x22=x12;
}
else{
k=(y12-yc2)/(x12-xc2);
k=atan(1/bk(k));
if(((x12-xc2)>0)&(&(y12-yc2)>0)){
  if(shape2==2){
    x22=x12+LL*sin(k);
    y22=y12-LL*cos(k);
  }
  if(shape2==3){
    x22=x12-LL*sin(k);
    y22=y12+LL*cos(k);
  }
}
if(((x12-xc2)<0)&(&(y12-yc2)>0)){
  if(shape2==2){
    x22=x12+LL*sin(k);
    y22=y12+LL*cos(k);
  }
  if(shape2==3){
    x22=x12-LL*sin(k);
    y22=y12-LL*cos(k);
  }
}
if(((x12-xc2)<0)&(&(y12-yc2)<0)){
  if(shape2==2){
    x22=x12-LL*sin(k);
    y22=y12+LL*cos(k);
  }
  if(shape2==3){
    x22=x12+LL*sin(k);
    y22=y12-LL*cos(k);
  }
}
if(((x12-xc2)>0)&(&(y12-yc2)<0)){
  if(shape2==2){
    x22=x12-LL*sin(k);
    y22=y12-LL*cos(k);
  }
  if(shape2==3){
    x22=x12+LL*sin(k);
    y22=y12+LL*cos(k);
  }
}
}
if((p1->type==ARC)&&(p2->type==LINE))
\[
\begin{align*}
\text{x12} &= \text{p2}->\text{graph.line.x_start}; \\
\text{y12} &= \text{p2}->\text{graph.line.y_start}; \\
\text{x22} &= \text{p2}->\text{graph.line.x_end}; \\
\text{y22} &= \text{p2}->\text{graph.line.y_end}; \\
\text{xc1} &= \text{p1}->\text{graph.arc.x_acenter}; \\
\text{yc1} &= \text{p1}->\text{graph.arc.y_acenter}; \\
\text{shape1} &= \text{p1}->\text{graph.arc.shape}; \\
\text{if}((\text{fabs}(\text{xc1}-\text{x12})<=0.001)\{ \\
&\quad \text{if}(\text{y12}>\text{yc1})\{ \\
&\quad\quad \text{x11}=\text{x12}-\text{LL}; \\
&\quad\quad \text{if}(\text{shape1}==2) \quad \text{y11}=\text{y12}; \\
&\quad\quad \text{else if}(\text{shape1}==3) \quad \text{y11}=\text{y12}+\text{LL}; \\
&\quad\} \\
&\quad \text{if}(\text{y12}<\text{yc1})\{ \\
&\quad\quad \text{x11}=\text{x12}+\text{LL}; \\
&\quad\quad \text{if}(\text{shape1}==2) \quad \text{y11}=\text{y12}; \\
&\quad\quad \text{else if}(\text{shape1}==3) \quad \text{y11}=\text{y12}-\text{LL}; \\
&\quad\} \\
&\} \\
&\text{if}(\text{fabs}(\text{x12}-\text{xc1})>0.001)\{ \\
&\quad \text{if}(\text{y12}==\text{yc1})\{ \\
&\quad\quad \text{x11}=\text{xc1}; \\
&\quad\quad \text{if}(\text{shape1}==2) \quad \text{y11}=\text{y12}+\text{LL}; \\
&\quad\quad \text{else if}(\text{shape1}==3) \quad \text{y11}=\text{y12}-\text{LL}; \\
&\quad\} \\
&\quad \text{if}(\text{x12}<\text{xc1})\{ \\
&\quad\quad \text{x11}=\text{x12}; \\
&\quad\quad \text{if}(\text{shape1}==2) \quad \text{y11}=\text{y12}+\text{LL}; \\
&\quad\quad \text{else if}(\text{shape1}==3) \quad \text{y11}=\text{y12}-\text{LL}; \\
&\quad\} \\
&\quad \text{else} \{ \\
&\quad\quad \text{k}=(\text{y12}-\text{yc1})/(\text{x12}-\text{xc1}); \\
&\quad\quad \text{k}=\text{atan}(\text{fabs}(\text{k})); \\
&\quad\quad \text{if}(((\text{x12}-\text{xc1})>0) && ((\text{y12}-\text{yc1})>0))\{ \\
&\quad\quad\quad \text{if}(\text{shape1}==3)\{ \\
&\quad\quad\quad\quad \text{x11}=\text{x12}+\text{LL}*\sin(\text{k}); \\
&\quad\quad\quad\quad \text{y11}=\text{y12}-\text{LL}*\cos(\text{k}); \\
&\quad\quad\quad\} \\
&\quad\quad\quad \text{else if}(\text{shape1}==2)\{ \\
&\quad\quad\quad\quad \text{x11}=\text{x12}-\text{LL}*\sin(\text{k}); \\
&\quad\quad\quad\quad \text{y11}=\text{y12}+\text{LL}*\cos(\text{k}); \\
&\quad\quad\quad\} \\
&\quad\quad\} \\
&\quad\quad \text{else if}(((\text{x12}-\text{xc1})<0) && ((\text{y12}-\text{yc1})>0))\{ \\
&\quad\quad\quad \text{if}(\text{shape1}==3)\{ \\
&\quad\quad\quad\quad \text{x11}=\text{x12}+\text{LL}*\sin(\text{k}); \\
&\quad\quad\quad\quad \text{y11}=\text{y12}+\text{LL}*\cos(\text{k}); \\
&\quad\quad\quad\} \\
&\quad\quad\quad \text{else if}(\text{shape1}==2)\{ \\
&\quad\quad\quad\quad \text{x11}=\text{x12}-\text{LL}*\sin(\text{k}); \\
&\quad\quad\quad\quad \text{y11}=\text{y12}-\text{LL}*\cos(\text{k}); \\
&\quad\quad\quad\} \\
&\quad\quad\} \\
&\quad\quad \text{else if}(((\text{x12}-\text{xc1})<0) && ((\text{y12}-\text{yc1})<0))\{ \\
&\quad\quad\quad \text{if}(\text{shape1}==3)\{ \\
&\quad\quad\quad\quad \text{x11}=\text{x12}-\text{LL}*\sin(\text{k}); \\
&\quad\quad\quad\quad \text{y11}=\text{y12}+\text{LL}*\cos(\text{k}); \\
&\quad\quad\quad\} \\
&\quad\quad\quad \text{else if}(\text{shape1}==2)\{ \\
&\quad\quad\quad\quad \text{x11}=\text{x12}+\text{LL}*\sin(\text{k}); \\
&\quad\quad\quad\quad \text{y11}=\text{y12}-\text{LL}*\cos(\text{k}); \\
&\quad\quad\quad\} \\
&\quad\quad\} \\
&\quad\} \\
&\} \\
\end{align*}
\]
if(shape1==3){
    x11=x12-LL*sin(k);
    y11=y12+LL*cos(k);
}
else if(shape1==2){
    x11=x12+LL*sin(k);
    y11=y12-LL*cos(k);
}
if((x12-xc1)>0)&(&((y12-yc1)<0)){
    if(shape1==3){
        x11=x12-LL*sin(k);
        y11=y12-LL*cos(k);
    }
    else if(shape1==2){
        x11=x12+LL*sin(k);
        y11=y12-LL*cos(k);
    }
}
}
num1=p1->num;
p0=(struct ELEMENTS *)malloc(sizeof(struct ELEMENTS));
if(p0){
    MessageBox(hWnd, "memory error!", MB_ICONSTOP|MB_OK);
    exit(0);
}
p1->next_e=p0;
p0->add=0;
p0->prior_e=p1;
p0->type=LINE;
// if(p1->type==LINE) p0->num-=1;
// else { p0->num=num1+1; num0++; }
p0->num=num1+1;
p0->graph_line.x_start=x12;
p0->graph_line.y_start=y12;
if(fabs(x12-x11)<0.001){
    if(y11<y12) y=y12+CL;
    if(y11>y12) y=y12-CL;
    x=x12;
}
else if(fabs(x12-x11)<0.001){
    if(y12==y11){
        if(x12>x11) x=x12+CL;
        if(x12<x11) x=x12-CL;
        y=y12;
    }
    else {
        k=(y12-y11)/(x12-x11);
        k=rnt(k);
        if(((x12-x11)>0)&(&((y12-y11)>0))
            { x=x12+CL*cos(k); y=y12+CL*sin(k); }
if(((x12-x11)<0)&(&(y12-y11)>0))
{
    x=x12-CL*cos(k);
    y=y12+CL*sin(k);
}
if(((x12-x11)<0)&(&(y12-y11)<0))
{
    x=x12-CL*cos(k);
    y=y12-CL*sin(k);
}
if(((x12-x11)>0)&(&(y12-y11)<0))
{
    x=x12+CL*cos(k);
    y=y12-CL*sin(k);
}

p0->graph.line.x_end=x;
p0->graph.line.y_end=y;
if(fabs(x12-x22)<0.001){
    if(y22<y12) y=y12+CL;
    if(y12<y22) y=y12-CL;
    x=x12;
}
if(fabs(x12-x22)>0.001){
    if(y12==y22){
        if(x12>x22) x=x12+CL;
        if(x12<x22) x=x12-CL;
        y=y12;
    } else {
        k=(y12-y22)/(x12-x22);
        k=atan(fabs(k));
        if(((x12-x22)>0)&(&(y12-y22)>0))
        {
            x=x12+CL*cos(k);
            y=y12+CL*sin(k);
        }
        if(((x12-x22)<0)&(&(y12-y22)>0))
        {
            x=x12-CL*cos(k);
            y=y12+CL*sin(k);
        }
        if(((x12-x22)<0)&(&(y12-y22)<0))
        {
            x=x12-CL*cos(k);
            y=y12-CL*sin(k);
        }
        if(((x12-x22)>0)&(&(y12-y22)<0))
        {
            x=x12+CL*cos(k);
            y=y12-CL*sin(k);
        }
    }
}
p3=(struct ELEMENTS *)malloc(sizeof(struct ELEMENTS));
if(!p3){
    MessageBox(hWnd, "memory error!", "", MB_ICONSTOP|MB_OK);
    exit(0);
}

p0->next_e=p3;
p3->prior_e=p0;
p3->type=LINE;
p3->add=0;
// num0++;
// if(p1->type==LINE) p3->num=num1+1;
// else p3->num=num1+2;
p3->num=num1+2;
p3->graph.line.x_start=p0->graph.line.x_end;
p3->graph.line.y_start=p0->graph.line.y_end;
p3->graph.line.x_end=x;
p3->graph.line.y_end=y;
p0=(struct ELEMENTS *)malloc(sizeof(struct ELEMENTS));
if(!p0){
    MessageBox(hWnd, "memory error!", "", MB_ICONSTOP|MB_OK);
    exit(0);
}

p3->next_e=p0;
p0->prior_e=p3;
p0->next_e=p2;
p2->prior_e=p0;
p0->type=LINE;
p0->add=0;
p0->graph.line.x_start=x;
p0->graph.line.y_start=y;
p0->graph.line.x_end=x12;
p0->graph.line.y_end=y12;
//if(p2->type==LINE) p0->num=-1;
// else { p0->num=num1+3; num0++;}
p0->num=num1+3;
p2->num=num1+4;
p0=p1->next_e;
// while(p0){
//     draw_elem(p0,8,4);
//     p0=p0->next_e;
//     if(p0==p2) break;
// }
info->members=info->members+3;
return p0;
}
*/

struct ELEMENTS *add_path(struct ELEMENTS *p1,struct RINGS *info)
{
struct ELEMENTS *p2,*p0,*p3;
float x1,y1,x2,y2,k1,k2,x0,y0;
float LL=16.0;
int num1,shape;

p2=p1->next_e;
if(p2==NULL) return p2;//p2=info->pElem;
if(p1->type==LINE)
{
Appendix 4

```c
xe=p1->graph.line.x_end;
ye=p1->graph.line.y_end;
xs=p1->graph.line.x_start;
ys=p1->graph.line.y_start;
k1=slope_angle(xs,ys,xs,xe);
}
if(p1->type==ARC)
{
    xe=p1->graph.arc.x_end;
ye=p1->graph.arc.y_end;
xc=p1->graph.arc.x_center;
yc=p1->graph.arc.y_center;
shape=p1->graph.arc.shape;
k1=slope_angle(xe,ye,xc,yc);
if(shape==2) k1=k1+Pi/2;
else k1=k1-Pi/2;
}
    x0=xe;
y0=ye;
x1=xe+LL*cos(k1);
y1=ye+LL*sin(k1);
if(p2->type==LINE)
{
    xe=p2->graph.line.x_end;
ye=p2->graph.line.y_end;
xs=p2->graph.line.x_start;
ys=p2->graph.line.y_start;
k2=slope_angle(xe,ye,xs,ys);
}
if(p2->type==ARC)
{
    xs=p2->graph.arc.x_start;
ys=p2->graph.arc.y_start;
xc=p2->graph.arc.x_center;
yc=p2->graph.arc.y_center;
shape=p2->graph.arc.shape;
k2=slope_angle(xs,ys,xc,yc);
if(shape==2) k2=k2+Pi/2;
else k2=k2-Pi/2;
}
    x2=xs+LL*cos(k2);
y2=ys+LL*sin(k2);

num1=p1->num;
p0=(struct ELEMENTS *)malloc(sizeof(struct ELEMENTS));
if(!p0)
    MessageBox(hWnd, "memory error!", ".", MB_ICONSTOP|MB_OK);
    exit(0);
}
p1->next_e=p0;
p0->add=0;
p0->prior_e=p1;
p0->type=LINE;
p0->graph.line.x_start=x0;
p0->graph.line.y_start=y0;
p0->graph.line.x_end=x1;
```

-195-
p0->graph->line.y_end=y1;
p3=(struct ELEMENTS *)malloc(sizeof(struct ELEMENTS));
if(!p3)
{
    MessageBox(hWnd, "memory error!", ",", MB_ICONSTOP|MB_OK);
    exit(0);
}
p0->next_e=p3;
p3->prior_e=p0;
p3->type=LINE;
p3->add=0;
p3->graph->line.x_start=p0->graph->line.x_end;
p3->graph->line.y_start=p0->graph->line.y_end;
p3->graph->line.x_end=x2;
p3->graph->line.y_end=y2;
p0=(struct ELEMENTS *)malloc(sizeof(struct ELEMENTS));
if(!p0)
{
    MessageBox(hWnd, "memory error!", ",", MB_ICONSTOP|MB_OK);
    exit(0);
}
p3->next_e=p0;
p0->prior_e=p3;
p0->next_e=p2;
p2->prior_e=p0;
p0->type=LINE;
p0->add=0;
p0->graph->line.x_start=x2;
p0->graph->line.y_start=y2;
p0->graph->line.x_end=x0;
p0->graph->line.y_end=y0;
p0=p1->next_e;
while(p0)
{
    num1++;
    p0->num=num1;
    p0=p0->next_e;
}
p0=p1->next_e;
info->members=info->members+3;
return p0;


REFERENCES

25-30.


REFERENCES


REFERENCES


-200-
REFERENCES


REFERENCES


ISO 10303-11 (1994E), Industrial Automation Systems and Integration-Product Data Representation and
REFERENCES


REFERENCES


REFERENCES


REFERENCES

37.


PrISMS (1996), NASA Program Information Systems Mission Services (PrISMS), Technical report, NASA.


REFERENCES


REFERENCES


REFERENCES

Air and Waste Management Association's 89th Annual Meeting and Exhibition, Navistar, TN, pp. 93-100.


REFERENCES


REFERENCES


VITA

Shane (Shengquan) XIE was born in Yiyang, Hunan province, P.R. China on January 21, 1970. He attended Yiyang No. 1 High School. He earned his Bachelor of Science degree in Automatic Control Engineering from Huazhong University (Wuhan, HuBei province, P. R. China) in 1992. He earned his Master of Science degree in Mechanical Engineering from Huazhong University of Science and Technology in 1994. He earned his first PhD degree of Engineering in the area of Mechatronics from Huazhong University of Science and Technology. His graduate work was supported by the National 863 High Technology Plan and National 85 Project of P.R. China. After that he worked as a research fellow at Department of Mechanical Engineering, University of Canterbury. At the same time, he earned his second PhD degree of engineering in the area of Manufacturing. His graduate work is greatly supported by the Public Good Science Funding of New Zealand. Upon graduation, he has accepted a position as a FRST (Foundation for Research, Science and Technology) postdoctoral fellow.