Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.

2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.

3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.
THE HONG KONG POLYTECHNIC UNIVERSITY

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

Generic methodology for the design and development of a responsive product development system

By

LEE Ka Man

A thesis submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

June 2004
CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

LEE Ka Man
Abstract

The product development environment is characterized by aggressive global competition, rapidly changing technologies and increasingly complex markets, all of which have prompted the development of knowledge-based systems to facilitate the exchange and updating of relevant product information. Decisions made during the early stage of the product development process are based on personal experience and knowledge. Knowledge based systems are normally used by enterprises at different stages of product development to enhance their efficiency in solving engineering problems. However, little attention has been given to replicate the data in the database system for supporting the formation of a knowledge repository. Traditionally, various functional disciplines have had their own database storage systems, using dissimilar data formats and even heterogeneous platforms. This creates problems in data exchange among enterprises due to the incompatibility of the data formats involved. Therefore, this research aims to develop a “responsive” product development system, with a combination of emerging technologies, to achieve the capturing and eventually the deployment of knowledge, thus facilitating effective data interchange among various computer systems.

In this research, the underlying technologies of a Responsive Product Development System (RPDS) encompass object technology, Case Based Reasoning (CBR) and Extensible Markup Language (XML), capitalizing on the unique features of the technologies involved. In brief, RPDS consists of three modules, which are the Object Based Product Development Module (OBPDM), the Product Information Exchange Module (PIEM) and the Hybrid Knowledge Module (HKM). OBPDM is used to model the product development process and the components of the process
with object technology. PIEM introduces a dynamic product information schema which is characterized by its ability to provide design practitioners with a product data exchange standard, thus enabling data interchange to take place among suppliers, customers and other business partners. The HKM provides a standardized mechanism of data exchange to facilitate dynamic data-knowledge conversion.

To validate the feasibility of the proposed schema, three case studies have been conducted in three local companies based on the suggested approach. Following from the feedback from these companies, a further review of the design of the proposed system was made in order to ensure the efficient information flow across the heterogeneous computer systems. To facilitate the deployment of RPDS, a generic methodology related to the design and implementation of the proposed system has been detailed with trial runs to validate its feasibility. The proposed methodology is a step in the development of a generic model for effective communication with business partners in order to achieve good collaboration, therefore attaining the goal of shortening the time-to-market and development of innovative products. Case based maintenance is attracting increasing attention and my research is about maintaining adaptable cases by deploying a "generalization-specification" method, which is a novel approach in the CBR cycle (Retrieve-Reuse-Revise-Retain). The significance of this research is the provision of a cross-platform data exchange system, which is able to facilitate the creation of an environment conducive to product development.
Publications Arising from the Thesis

(7 journal papers are published or accepted and 1 journal paper is under review.
6 conference papers are published or under reviewed)


Acknowledgements

I would like to express my thanks to my chief supervisor, Dr. H.C.W. Lau for his valuable advice, insightful comments, and innovative suggestions on my research work over the entire research period. Sincere thanks also to my co-supervisor, Dr. K. M. Yu for providing the research direction and encouraging independent thinking in my professional growth. I would like to thank Mr. Peter Jackson for proofreading the dissertation. I would like to express my gratitude to my colleague in the Department of Industrial and Systems Engineering for their thoughtful discussions and assistance.

I would like to thank people associated with Group Sense Limited, Gara Plastic Products Factory Limited and Confi Metal Finishing Company Limited for providing their cases of product development. I express my appreciation for Miss Elizabeth Nam, Mr. Matthew Chong, Miss Brenda Mok, Mr. Yannis So, Mr. Lee Chung and Miss Judy Sung for providing me with support throughout my case study, in their companies.

I am grateful to my family for their love and wholehearted support and also Mr. Byron Au for his invaluable suggestions and continuous encouragement.

The research described in this thesis is supported by a grant from the Research Committee of the Hong Kong Polytechnic University (No G-W 065).
# TABLE OF CONTENTS

ABSTRACT .................................................................................................................. i  
PUBLICATIONS ARISING FROM THE THESIS ...................................................... iii 
ACKNOWLEDGEMENTS ......................................................................................... v 
TABLES OF CONTENTS ......................................................................................... vi 
LIST OF FIGURES ................................................................................................... xi 
LIST OF TABLES ....................................................................................................... xiv 

## CHAPTER 1. INTRODUCTION ................................................................................. 1.1  
1.1 Research Background .................................................................................... 1.1 
1.2 Statement of Problem .................................................................................... 1.2 
1.3 Research Objective ....................................................................................... 1.4 
1.4 Significance of the Research ......................................................................... 1.4 
1.5 Organization of the Dissertation .................................................................... 1.5 

## CHAPTER 2. LITERATURE REVIEW AND INDUSTRIAL SURVEY ....................... 2.1  
2.1 Academic Study of Information Systems for Product Development ............... 2.2  
2.1.1 The Evolution of Information Systems for Product Development ............ 2.2  
2.1.2 Research on Object Technology, Case Based Reasoning and XML .......... 2.7  
2.1.2.1 Product Data Management (PDM) System Modeling with OO Approach ................................................................. 2.8  
2.1.2.2 Knowledge Based System for Product Development ...................... 2.9  
2.1.2.3 Existing Product Data Exchange Standard ..................................... 2.14  
2.1.2.4 Review of XML Adoption in Industry ........................................... 2.20 
2.2 An Exploratory Study in Adoption of Information Systems for Product Development ................................................................. 2.22 
2.2.1 Related Studies on Information Systems for Managing Product Lifecycle .............................................................................................................. 2.23 
2.2.2 Design of Questionnaire and Data Collection ........................................ 2.25 
2.2.3 Observation ............................................................................................... 2.27 
2.2.4 Discussion ................................................................................................. 2.32 
2.3 Implications from the Literature Review ....................................................... 2.33
# Table of Contents

## Chapter 3. Design of a Responsive Product Development System (RPDS)

3.1 Features of RPDS ......................................................... 3.2
   3.1.1 Agility ................................................................. 3.2
   3.1.2 Self-learning Ability ............................................... 3.3
   3.1.3 Customizability .................................................... 3.5

3.2 Underlying Technologies for RPDS ...................................... 3.5

3.3 System Architecture of RPDS ............................................ 3.7

3.4 Design Considerations ................................................... 3.8

3.5 Modules of RPDS .......................................................... 3.10
   3.5.1 Object Based Product Development Module (OBPDM) ....... 3.11
   3.5.2 Hybrid Knowledge Module (HKM) .......................... 3.12
   3.5.3 Product Information Exchange Module (PIEM) .............. 3.12

## Chapter 4. Development of Object Based Product Development Module (OBPDM)

4.1 Concept of Object-Oriented Design and Analysis ................. 4.2

4.2 Object-Oriented Modeling .............................................. 4.5
   4.2.1 Static Structural Modeling ..................................... 4.6
   4.2.2 Dynamic Modeling ............................................... 4.10

4.3 Timely Reconfigurable Method Invocation .......................... 4.14
   4.3.1 Declaration of TRMI ............................................. 4.15
   4.3.2 The Logic of TRMI ............................................... 4.16
   4.3.3 Static Representation of TRMI ............................... 4.18
   4.3.4 Dynamic Modeling of TRMI .................................... 4.20
   4.3.5 Case Example in Product Development Process .......... 4.21

## Chapter 5. Development of Hybrid Knowledge Module (HKM)

5.1 Mapping between the Corporate Database and the Case Base .... 5.1
   5.1.1 Corporate Database ............................................... 5.2
   5.1.2 Knowledge Repository .......................................... 5.4

5.1.3 The Dynamic Data Exchange Mechanism between Knowledge Repository and Relational Database .......................... 5.5

5.2 Objectization of Case Based Reasoning .............................. 5.9
5.2.1 Case Representation ...................................................................5.10
  5.2.1.1 Generalizing the Case (Objects) to Form an Abstract Case
           (Abstract Class) or a Generalized Case (Superclass) ...............5.11
  5.2.1.2 Case Categorization Based on the K-Nearest Neighbour Approach.
           .....................................................................................5.11
  5.2.1.3 Grouping ...........................................................................5.13
  5.2.1.4 Representative Name for the Generalized Case ......................5.14

5.2.2 Case Retrieval ...........................................................................5.15
  5.2.2.1 Case Selection .....................................................................5.15
  5.2.2.2 Inductive Retrieval .............................................................5.15
  5.2.2.3 Comparison by Paying Attention to the Semantics of the Case...5.17

5.2.3 Case Adaptation .........................................................................5.19
  5.2.3.1 Backward Chaining from the Target Object to Superclass ........5.19
  5.2.3.2 Deontic Logic in Case Adaptation ........................................5.21

5.2.4 Case Maintenance .....................................................................5.25

CHAPTER 6. DEVELOPMENT OF PRODUCT INFORMATION
EXCHANGE MODULE (PIEM) .......................................................6.1

6.1 Need for Product Information Standard .....................6.3
6.2 Features of PIML in PIEM .................................................6.5
6.3 Development of PIEM .......................................................6.5
  6.3.1 Create Product Development Model .................................6.6
  6.3.2 Convert it into an Object Model ......................................6.7
  6.3.3 Define XML Schema Based on Object Model .................6.7
    6.3.3.1 Association .....................................................................6.8
    6.3.3.2 Inheritance ....................................................................6.10
    6.3.3.3 Polymorphism ...............................................................6.12

6.4 Applications of PIEM ..........................................................6.14
  6.4.1 Data Transformation to a Knowledge Base ..................6.15
    6.4.1.1 Product Data Schema .................................................6.16
    6.4.1.2 Mapping Mechanism Between a Relational Database and a
            Knowledge Repository .....................................................6.18
  6.4.2 Web-Based Support for Product Data Management .......6.21

CHAPTER 7. IMPLEMENTATION OF RPDS .........................7.1
7.1 Project Start Up Stage .........................................................7.2
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1.1 Technical Level</td>
<td>7.2</td>
</tr>
<tr>
<td>7.1.2 Organizational Level</td>
<td>7.4</td>
</tr>
<tr>
<td>7.2 Structural Formulation Stage</td>
<td>7.6</td>
</tr>
<tr>
<td>7.2.1 OBPDM—The Requirement Analysis for Object Oriented Application and Object-Oriented Design</td>
<td>7.6</td>
</tr>
<tr>
<td>7.2.2 HKM—Building the CBR System</td>
<td>7.9</td>
</tr>
<tr>
<td>7.2.2.1 Defining the Case Structure</td>
<td>7.9</td>
</tr>
<tr>
<td>7.2.2.2 Authorizing the Case</td>
<td>7.9</td>
</tr>
<tr>
<td>7.2.2.3 Study of the Retrieval Mechanism and Setting Questions for Cases</td>
<td>7.11</td>
</tr>
<tr>
<td>7.2.2.4 Reusing the Knowledge by Derivational Adaptation</td>
<td>7.12</td>
</tr>
<tr>
<td>7.2.3 PIEM—Semantics of Product Information Markup Language</td>
<td>7.12</td>
</tr>
<tr>
<td>7.3 Evaluation and Testing Stage</td>
<td>7.14</td>
</tr>
<tr>
<td>7.3.1 OBPDM—Test for Agility</td>
<td>7.15</td>
</tr>
<tr>
<td>7.3.2 HKM—Test for Self-learning</td>
<td>7.17</td>
</tr>
<tr>
<td>7.3.3 PIEM—Test for Customizability</td>
<td>7.20</td>
</tr>
<tr>
<td>7.4 Operational Stage</td>
<td>7.22</td>
</tr>
<tr>
<td>7.4.1 Technical Aspect of RPDS</td>
<td>7.22</td>
</tr>
<tr>
<td>7.4.2 Management Aspect of RPDS</td>
<td>7.24</td>
</tr>
</tbody>
</table>

CHAPTER 8. CASE STUDY ................................................................. 8.1

8.1 Case Study in Group Sense Limited ........................................ 8.1

8.1.1 Existing Workflow and Challenges Faced by GSL ....................... 8.2

8.1.2 Roadmap for Implementing RPDS .......................................... 8.4

8.1.2.1 Analysis of the Data Structure with an Object Oriented Concept .. 8.5

8.1.2.2 Simplify the Data Structure by Inheritance ........................ 8.7

8.1.2.3 Construct the Transformation Rule for Information Manipulation ........................................... 8.9

8.1.2.4 Data Conversion by PiML to Knowledge Base ........................ 8.13

8.1.2.5 Application of CBR in RPDS ........................................... 8.15

8.2 Case Study in Gara Plastic Products Factory Limited ..... 8.18

8.2.1 Existing Workflow and Challenges Faced by Gara ...................... 8.19

8.2.2 A Roadmap for Implementing RPDS ....................................... 8.20

8.2.2.1 Construction of Object Model in OOPDM ............................... 8.21

8.2.2.2 Transformation of Knowledge from Database in HKM ................ 8.23

8.2.2.3 Deployment of PiML in PDEM ........................................... 8.27

8.3 Case Study in Confi Metal Finishing Company Limited... 8.36

8.3.1 Existing Workflow and Challenges Faced by Confi ...................... 8.36

8.3.2 A Roadmap for Implementing RPDS ....................................... 8.38
**Table of Contents**

CHAPTER 9. DISCUSSION .................................................................................. 9.1  
  9.1 General Discussion .............................................................................. 9.1  
  9.2 Discussion of OBPDM ......................................................................... 9.2  
  9.3 Discussion of HKM ............................................................................. 9.3  
  9.4 Discussion of PIEM ............................................................................. 9.4  
  9.5 Discussion of the Case Study ................................................................. 9.5  

CHAPTER 10. CONCLUSION ......................................................................... 10.1  
  10.1 Summary of Research Work ................................................................. 10.1  
  10.2 Contribution of the Research ................................................................. 10.2  
  10.3 Future Research ................................................................................. 10.4  

APPENDICS  
Appendix A - Generic specification for product design--Product  
  Information Markup Language ................................................................. A.1  
Appendix B - Product Development Survey ............................................... B.1  
Appendix C - The Measurement of Using Information System to Support  
  Product Development ............................................................................... C.1  
Appendix D - Documents from Case Companies ......................................... D.1  
Appendix E - Project Report for Gara Plastic Products Factory Limited  
  .................................................................................................................. E.1  

References ................................................................................................. R.1
LIST OF FIGURES

Figure 3.1 The architecture of RPDS ................................................................. 3.11
Figure 4.1 Many parts are part of one product or one case consists of solution and
    problem ........................................................................................................... 4.8
Figure 4.2 Sequence diagram of engineering change management ...................... 4.14
Figure 4.3 Process flow of TRMI for product development ............................... 4.17
Figure 4.4 Object model to illustrate the concept of polymorphism.................... 4.19
Figure 4.5 State transition of CAD modification .............................................. 4.21
Figure 4.6 Object model of dynamic objects .................................................... 4.23
Figure 5.1 Correlation between case base and relational database .................... 5.3
Figure 5.2 The flow of updating the case base with corporate data .................... 5.6
Figure 5.3 Steps for objectization of Case Based Reasoning ............................. 5.10
Figure 5.4 Depths of the decision tree ............................................................ 5.14
Figure 5.5 The generalized decision tree ......................................................... 5.16
Figure 5.6 Generalizing the case to form a superclass and deriving a new solution by
    specification .................................................................................................... 5.20
Figure 6.1 The framework of PIEM ................................................................. 6.3
Figure 6.2 The major steps for objectization .................................................... 6.6
Figure 6.3 XSD and UML class diagram ........................................................ 6.8
Figure 6.4 Express associations in XSD ............................................................ 6.9
Figure 6.5 Generalization and specification ...................................................... 6.10
Figure 6.6 Express inheritance with XSD ......................................................... 6.11
Figure 6.7 XSD of engineering change request ............................................... 6.18
Figure 6.8 XML of engineering change request .............................................. 6.19
Figure 6.9 The neutral data file for the case base .............................................. 6.20
Figure 6.10 A snapshot of PIIM ....................................................................... 6.22
Figure 7.1 Generic workflow of product development ...................................... 7.14
Figure 7.2 Schema overview of customer requests ......................................... 7.21
Figure 7.3 A brief message of successful validation ......................................... 7.21
Figure 7.4 A brief message of unsuccessful validation ..................................... 7.21
Figure 8.1 Workflow of GSL ........................................................................... 8.3
Figure 8.2 The key steps of data analysis and transformation .......................... 8.5
Figure 8.3 Object oriented model of product development in GSL ................. 8.6
Figure 8.4 Snapshot of the screen of data conversion ....................................... 8.14
Figure 8.5 Case retrieval module of CBR ....................................................... 8.17
Figure 8.6 Sequence diagram of product development in Gara ...................... 8.20
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.7</td>
<td>The object model of Gara</td>
</tr>
<tr>
<td>8.8</td>
<td>Problem resolution of RPDS</td>
</tr>
<tr>
<td>8.9</td>
<td>CAD Library within the Case Base Reasoning system</td>
</tr>
<tr>
<td>8.10</td>
<td>Co-relation between the case base and the relational database</td>
</tr>
<tr>
<td>8.11</td>
<td>Constructing XSD with XML Spy</td>
</tr>
<tr>
<td>8.12</td>
<td>Validate the XML file with XML schema</td>
</tr>
<tr>
<td>8.13</td>
<td>The interface of the Design Drawing Start Form</td>
</tr>
<tr>
<td>8.14</td>
<td>HTML is formed by serializing the XML file with XSLT</td>
</tr>
<tr>
<td>8.15</td>
<td>Production Workflow at Confi</td>
</tr>
<tr>
<td>8.16</td>
<td>Snapshot of web-based CBR</td>
</tr>
<tr>
<td>A.1</td>
<td>Generic workflow of product design</td>
</tr>
<tr>
<td>A.2</td>
<td>Schema overview of Project Mission Statement</td>
</tr>
<tr>
<td>A.3</td>
<td>Schema overview of customer expectations</td>
</tr>
<tr>
<td>A.4</td>
<td>Schema overview of product benchmarking</td>
</tr>
<tr>
<td>A.5</td>
<td>Schema overview of design specification</td>
</tr>
<tr>
<td>A.6</td>
<td>Schema overview of functional decomposition</td>
</tr>
<tr>
<td>A.7</td>
<td>Schema overview of concept generation</td>
</tr>
<tr>
<td>A.8</td>
<td>Schema overview of concept evaluation</td>
</tr>
<tr>
<td>A.9</td>
<td>Schema overview of cost sheet</td>
</tr>
<tr>
<td>A.10</td>
<td>Schema overview of prototyping</td>
</tr>
<tr>
<td>C.1</td>
<td>Common problem found in product development</td>
</tr>
<tr>
<td>C.2</td>
<td>The method of problem solving adopted by companies</td>
</tr>
<tr>
<td>C.3</td>
<td>Design change encountered by companies</td>
</tr>
<tr>
<td>C.4</td>
<td>Knowledge management adopted by companies</td>
</tr>
<tr>
<td>C.5</td>
<td>Function of information system</td>
</tr>
<tr>
<td>C.6</td>
<td>The proportion of PLM system self-developed or off-the shelf</td>
</tr>
<tr>
<td>C.7</td>
<td>Aspect of OO deployment</td>
</tr>
<tr>
<td>C.8</td>
<td>AI used by companies</td>
</tr>
<tr>
<td>C.9</td>
<td>Method of information exchange with business partners</td>
</tr>
<tr>
<td>C.10</td>
<td>Method of solving data exchange difficulties</td>
</tr>
<tr>
<td>C.11</td>
<td>Method of resolving the problem of data exchange</td>
</tr>
<tr>
<td>C.12</td>
<td>Industrial sector of the respondents</td>
</tr>
<tr>
<td>C.13</td>
<td>Position of the respondents</td>
</tr>
<tr>
<td>E.1</td>
<td>Design workflow</td>
</tr>
<tr>
<td>E.2</td>
<td>Prototype making workflow</td>
</tr>
<tr>
<td>E.3</td>
<td>The tool making workflow</td>
</tr>
<tr>
<td>E.4</td>
<td>Overview of OLAP</td>
</tr>
<tr>
<td>E.5</td>
<td>OLAP Excel PivotTable</td>
</tr>
<tr>
<td>E.6</td>
<td>OLAP Excel chart</td>
</tr>
<tr>
<td>E.7</td>
<td>OLAP query</td>
</tr>
<tr>
<td>E.8</td>
<td>Data transformation services</td>
</tr>
</tbody>
</table>
Figure E.9 CBR cycle ................................................................. E.26
Figure E.10 The interface of Problem Resolution of Case ADVISOR 4.12 ...... E.29
Figure E.11 The interface of Problem Resolution of Case ADVISOR 4.12 ...... E.30
Figure E.12 The interface of Problem Resolution of Case ADVISOR 4.12 ...... E.30
Figure E.13 The interface of Case Authoring of Case ADVISOR 4.12 .......... E.31
Figure E.14 The input interface for new domain .................................. E.32
Figure E.15 The interface of Decision Tree Editor .............................. E.32
Figure E.16 The interface of Edit Tree ......................................... E.33
Figure E.17 The interface for creating decision tree .......................... E.33
Figure E.18 The interface for creating decision tree .......................... E.34
Figure E.19 The interface for creating decision tree .......................... E.34
Figure E.20 The interface of Case Authoring.................................... E.35
Figure E.21 The interface of Edit Case ....................................... E.36
Figure E.22 The interface of Case Options .................................... E.37
Figure E.23 The interface of Case Authoring.................................... E.38
Figure E.24 The interface of Associate Question ............................. E.39
Figure E.25 The interface of Case Authoring.................................... E.39
Figure E.26 XML with XML Spy ............................................... E.43
Figure E.27 XML opened with web browser .................................... E.44
Figure E.28 HTML is formed by serializing the XML file with XSLT ......... E.45
Figure E.29 Connecting to web server ....................................... E.46
Figure E.30 XML is opened with spreadsheet .................................. E.46
Figure E.31 Data analysis with pivot table .................................... E.47
Figure E.32 Snapshot of data import ......................................... E.48
Figure E.33 DTS Import Wizard ............................................... E.48
Figure E.34 Destination for data import ..................................... E.49
Figure E.35 Source table selection ............................................ E.49
Figure E.36 Executing package for data import/export ....................... E.50
Figure E.37 SQL Server connection ......................................... E.50
Figure E.38 Data connection wizard ......................................... E.51
Figure E.39 Dragging the data field to ASP .................................. E.52
Figure E.40 Snapshot of IIS ................................................... E.53
Figure E.41 Setting the new virtual directory ................................ E.53
Figure E.42 Accessing data in SQL server through a browser ................ E.54
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>CBR project related to product design</td>
<td>2.11</td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison of exiting standards for product data exchange</td>
<td>2.19</td>
</tr>
<tr>
<td>2.3</td>
<td>Studies related to the application of object technology</td>
<td>2.24</td>
</tr>
<tr>
<td>2.4</td>
<td>The co-relation between design changes and the same design change</td>
<td>2.27</td>
</tr>
<tr>
<td>2.5</td>
<td>Knowledge Management (KM) methodology adopted by PRD manufacturers</td>
<td>2.28</td>
</tr>
<tr>
<td>2.6</td>
<td>The problems encountered by the respondents</td>
<td>2.30</td>
</tr>
<tr>
<td>2.7</td>
<td>Solutions to data exchange difficulties</td>
<td>2.30</td>
</tr>
<tr>
<td>2.8</td>
<td>Mean scores, rank and standard deviation of the benefits of data exchange</td>
<td>2.31</td>
</tr>
<tr>
<td>2.9</td>
<td>Mean scores, rank and standard deviation of the potential features</td>
<td>2.31</td>
</tr>
<tr>
<td>5.1</td>
<td>Product design cases extracted from the case base</td>
<td>5.16</td>
</tr>
<tr>
<td>6.1</td>
<td>Major elements of engineering change request</td>
<td>6.17</td>
</tr>
<tr>
<td>7.1</td>
<td>The roadmap for implementing RPDS</td>
<td>7.1</td>
</tr>
<tr>
<td>7.2</td>
<td>The technical aspect of implementing RPDS</td>
<td>7.4</td>
</tr>
<tr>
<td>7.3</td>
<td>Roles and responsibilities</td>
<td>7.5</td>
</tr>
<tr>
<td>8.1</td>
<td>Content of a case in HKM</td>
<td>8.15</td>
</tr>
<tr>
<td>8.2</td>
<td>Data mapping between corporate database to knowledge repositories</td>
<td>8.27</td>
</tr>
<tr>
<td>8.3</td>
<td>Process of migrating data from database to case base</td>
<td>8.35</td>
</tr>
<tr>
<td>A.1</td>
<td>Elements of project mission statement</td>
<td>A.3</td>
</tr>
<tr>
<td>A.2</td>
<td>Elements of customer expectations</td>
<td>A.5</td>
</tr>
<tr>
<td>A.3</td>
<td>Elements of product benchmarking</td>
<td>A.7</td>
</tr>
<tr>
<td>A.4</td>
<td>Elements of design specification</td>
<td>A.9</td>
</tr>
<tr>
<td>A.5</td>
<td>Elements of functional decomposition</td>
<td>A.13</td>
</tr>
<tr>
<td>A.6</td>
<td>Elements of concept generation</td>
<td>A.15</td>
</tr>
<tr>
<td>A.7</td>
<td>Elements of concept evaluation</td>
<td>A.17</td>
</tr>
<tr>
<td>A.8</td>
<td>Elements of cost sheet</td>
<td>A.20</td>
</tr>
<tr>
<td>A.9</td>
<td>Elements of prototyping</td>
<td>A.23</td>
</tr>
<tr>
<td>C.1</td>
<td>The benefits of data exchange</td>
<td>C.7</td>
</tr>
<tr>
<td>C.2</td>
<td>Potential feature of information system</td>
<td>C.8</td>
</tr>
<tr>
<td>C.3</td>
<td>Number of employee in non-manufacture sectors</td>
<td>C.8</td>
</tr>
<tr>
<td>E.1</td>
<td>Hong Kong's exports of toys</td>
<td>E.5</td>
</tr>
<tr>
<td>E.2</td>
<td>Hong Kong's total exports of toys by major market</td>
<td>E.6</td>
</tr>
<tr>
<td>E.3</td>
<td>Hong Kong's total exports of toys by major category</td>
<td>E.6</td>
</tr>
</tbody>
</table>
CHAPTER 1. INTRODUCTION

1.1 Research Background

In to-day's increasingly competitive environment, companies are experiencing growing pressures to reduce the product development lead-time to meet market expectations. At the same time, customers are asking for more varieties of product ranges, which preferably can be customized to suit their ever-changing requirements. In general, the earlier a product is introduced to the market, the greater the chance the company will be able to gain a better market share. On the other hand, if new products, especially "high tech" products, reach the market late, they will earn less profit. This relationship can clearly be illustrated by a bell curve (Prasad, 1996). Most importantly, there is an urgent need for all functional disciplines of companies to contribute to the development of products, whether it is an upgrade of an existing product or a brand-new concept of a novel product. This is particularly true in the case of New Product Development (NPD), which is considered as an activity, requiring the input and concerted effort of relevant functional disciplines covering all stages of product development, i.e. from conceptual design to practical manufacturing of the product. NPD is recognized as a crucial activity to most companies for securing long-term survival and growth (Brown and Eisenhardt, 1995).

Traditionally, NPD goes through various stages of sequential operations, causing inflexibility to cope with any unexpected changes such as the change which occurs when material fails to pass certain structural tests. NPD is also seen as a highly complex, uncertain and probabilistic process of collective action (Van de Ven et al., 1989). Moreover, the lack of a common format for information exchange among various disciplines aggravates the problem in terms of lead-time control and
communication of inter-related data. In particular, it is essential that the information related to the progress of NPD in various disciplines can be accessed and changed (by authorized personnel only) in an integrated manner. The experience and knowledge of product design as well as related workflow are conventionally kept in various departments, leading to scattered data storage areas. Thus, it will be a great step forward if a system is built to capture, in real time, the progress of various NPD activities, allowing an integrated view of a company's scattered knowledge sources.

Today's product development environment is characterized by aggressive global competition, rapidly changing technologies and increasingly complex markets (Floyd, 1993), all of which have prompted the development of knowledge-based systems which facilitate the exchange and updating of relevant product information. However, little attention has been paid to replicating the data in database systems to a knowledge repository. This prevents users from obtaining the latest information. Traditionally, various functional disciplines have had their own database storage systems, using dissimilar data formats and even heterogeneous platforms. This creates problems in data exchange among enterprises due to incompatibility of the data formats involved. Therefore, this research aims to develop a "Responsive" product development system with a combination of existing technologies to achieve the capturing and optimum deployment of knowledge.

### 1.2 Statement of Problem

Responsive product development is characterized by the early involvement of different functional disciplines, some of which may not have acquired the needed information due to various reasons such as lack of communication and understanding. As a result, decisions made in the early design phases of product development are
often based on poorly structured, immature and outdated information (Haque et al, 2000). Very often, decisions are based on personal experience and knowledge without any systematic or structured information support.

Knowledge based systems are comprehensively used by enterprises in product development life cycles to enhance their efficiency in solving the engineering problems. However, little concern is put on replicating the data in database system to a knowledge repository. That leads to non-synchronization of both storage systems and may greatly disrupt the work in that knowledge workers do not get the latest information on which to base decision.

At different stages of product development, decision-making is required so that product can be developed to confirm with various constraints and a range of variables. The pre-requisite for survival is being adaptive to the environment where the environment is full of unpredictable variables. Responsive methods in terms of managing and implementing changes are urgently needed. The information system is the backbone of an organization but the traditional static system model is insufficiently dynamic.

Analyzing, extracting, and finally supplying up-to-date data are the stepping stones to developing a responsive product development system. A corporation, whose business partners and globally-distributed customers have their own database storage system using dissimilar data formats and even heterogeneous platforms, find it difficult to collaborate with them.
1.3 Research Objective

The specific objective of this research is to:

i. Develop a "Responsive" product development system with a combination of existing technologies to achieve the capturing and optimum deployment of knowledge

ii. Improve the speed and effectiveness of the product development process

iii. Develop an information system model for Product Lifecycle Management (PLM) which is the extended concept of Product Data Management (PDM)

iv. Enhance the full capability of proprietary product design tools, with the support of universal data exchange among various platforms

1.4 Significance of the Research

My proposed methodology is a step in the development of a generic model for effective communication with business partners in order to achieve collaboration, therefore attaining the goals of shortening the time-to-market and developing innovative products. In particular, a universal product information exchange mechanism enables knowledge workers to acquire sufficient and accurate information. This research introduces object technology to develop RPDS such that the reusable and inter-related objects can respond to the turbulent market quickly by inheriting the common attribute values from the "parent" objects. In addition, the details of Case Based Reasoning (CBR) and the applications of machine learning in a product design environment have been studied. Case based maintenance is attracting increasing attention and my research is about maintaining adaptable cases by deploying a "generalization-specification" method, which is a novel approach in the CBR cycle (Retrieve-Reuse-Revise-Retain).
1.5 Organization of the Dissertation

The dissertation consists of ten chapters. The outline of the dissertation is as follows:

i. Chapter I states the problems that occur in product development and describes the motivation for and background to the research.

ii. Chapter II is an academic review of research on information systems on product development. In the second section of Chapter II, an industrial survey is used to explore the existing practice of product development.

iii. Chapter III depicts the features of Responsive Product Development System (RPDS) and technologies used in RPDS. It presents the system architecture of RPDS which consists of Object Based Product Development Module (OBPDM), Hybrid Knowledge Module (HKM) and Product Information Exchange Module (PIEM). The design considerations behind it are discussed.

iv. Chapter IV addresses the deployment of object technology in system modeling and specifies the timely reconfigurable method in product development activities.

v. Chapter V introduces a dynamic data exchange mechanism which attempts to exchange data automatically. The filtering of valuable data between traditional relational database model and CBR knowledge repository is also discussed.

vi. Chapter VI focuses on a pilot application of resolving conflict in data exchange by the use of the proposed Product Information Markup Language (PIML). PIML is a specific set of Extensible Markup Language (XML) to describe product data carried within the product development lifecycle.

vii. Chapter VII provides the generic implementation guide of RPDS from the
startup stage, though the structural formulation stage, evaluation and testing stages, to the operation stage.

viii. Chapter VIII is about conducting a case study among Hong Kong-based manufacturers. Once the workflow has been realized, the roadmap of implementing RPDS is constructed and customized processes for each firm are introduced.

ix. Chapter IX discusses the results of applying the architecture to case studies; and the deployment of RPDS.

x. Chapter X is the conclusion which presents the contributions of this study and also suggests recommendations for future research.
CHAPTER 2. LITERATURE REVIEW

Design processes evolve from new product introduction or incremental product improvement. The iterative product process involves a multidisciplinary design team in a distributed manufacturing environment, resulting in complex coordination. As design processes are interdependent or independent, the tasks are implemented concurrently or sequentially. Product design problems are usually sub-divided into sub-tasks and sub-problems. It is essential to synthesize the solution of the sub-problem and optimize the functionality of the product development operation. Since product development processes change according to the market trend and engineering constraints, design practitioners need to respond quickly to changes and uncertainties. They also need to evaluate all possible alternative solutions until the problems are resolved. In the past, most design tasks were done manually using the capability of collaboration, creativity and fast analytical thinking. To optimize both personal and process productivity, some researchers attempted to represent knowledge with artificial intelligence techniques to solve the problems encountered in product development activities.

This chapter aims to provide a comprehensive survey of product development as revealed in academic publications and existing practices of enterprises. The first section is a literature review conducted on product lifecycle management. Various technologies including artificial intelligence, object technology and the data exchange standard, have been reviewed. The other section is a survey conducted with the objective of exploring how information systems are used to assist knowledge workers during product development.
2.1 Academic Study of Information Systems for Product Development

This section reviews the past academic researches on information systems for product development. The first part of this section provides a general review on Computer Aided Design (CAD) or Computer Aided Manufacture (CAM) and expert systems for empowering the product development team; enhancing the productivity of labor and improving the utilization of resources from 70’s to 21st century. It is not difficult to find that there has been a significant improvement in the related technology of computers, graphic equipment, data retrieval and storage system and telecommunication in past 30 years. The second part of this section focuses on three particular emerging technologies which are: object technology for system modeling, case based reasoning for knowledge acquisition and sharing and XML for data exchange among heterogeneous systems and platforms.

2.1.1 The Evolution of Information Systems for Product Development

Industrial production is the process of converting raw materials into products which have a value for exchange which is greater than the cost (Archer, 1969). Before a decision is taken on whether to introduce a new product demand, cost, and profit and market uncertainty need to be considered carefully, in order to arrive at the optimal solution (Urban, 1968). Technology invention and aesthetic invention are major value-added elements for developing innovative product. Innovation was recognized in 60s; the obstacles to innovation are the lack of market feedback, lack of financial support and lack of liaison with industry (de Vitry, 1969). Total product development (Bennet, 1969) is suggested for the joint development of new ideas
among technical people, engineers and commercial people, to test the ideas with factual data and quantitative studies rather than guessing, to develop programs with detailed planning and to review the progress constantly by detecting changes in the market. In the early 60s computers were not commonly used and they were usually used for data manipulation and for documentation. Only renowned aerospace companies and motor companies (Warshawsky, 1965; Wilson, 1969; Matthews, 1969) started to make use of computers to calculate the significant design parameters of an airframe. Computer systems in the mid 60's could (i) perform analytical functions; (ii) design with rules supplied by production engineering organization; (iii) optionally choose cutters and feed rate to generate control tapes for numerical control manufacturing of limited tools, and produce inspection data (Wilson, 1969; Stephenson, 1969). However, computer systems were not yet able to alternate machine tool controlling information in real time and switch the work piece from one machine to another by detecting bread-down or overload (Wilson, 1969). Communication, operation and time taken in planning and processing were regarded as three major areas for automating design process (Warshawsky, 1965; Warwick, 1969).

In the late 60s and early 70s, CAD technology was advocated and CAD was used to represent shape, produce visualization of concepts to realize the effect of proposed products (Allan, 1972). The computer network by means of telephone lines and software tools enabled people to utilize computers to improve their work. The databases associated with CAD systems helped minimize the effort needed for updating files and performing information processing operations. CAM system was used for generating exact paths for guiding an object through exactly specified infinite positions. Design automation made use of CAD technology is a way to leverage
design and production performance (Adshead, 1962; Allen, 1974). However, the system in 70s was not aimed at integrated design and production (Gott, 1972). The product development process can be improved by identifying the capability of the firm, realizing the needs of customers and providing a vehicle to get information and foster an atmosphere of good communication within the firm (Phelps, 1977). A computer based system was used for the digital processing of dynamic strain data by an aircraft company; planning systems with actual cost and manpower data as input to produce a budget, cost estimates and manpower plans (Clark, 1976). Some Computer-based Management Systems which were used for cost-benefit analysis in the production and managerial spheres, showed how economic advantages could be obtained. (Surguchev, 1979; Drake, 1974)

In the early 80’s, information was scare and incomplete so industrial product development especially for new product development was regarded as risky and complex (CardozoJerry Wind, 1985). Urban and Hauser (1980) collected statistics to estimate the overall probability of success for the design and introduction of new consumer products to be 0.19. Innovation and marketing information is essential for generating concept for new products. Binetti reported that there was a new connection with the Lockheed Missiles and Space information Service, Palo Alto that could interrogate the database for online-information retrieval and update information thus helping design practitioners to develop creative ideas for new product development (Binetti, 1980). A fuzzy system was used as a decision support system to interpret the fuzzy information in new product development (Nojiri, 1982). However, the decision support system with AI technique was presented as a mathematical model and decision support system was based on theory rather than on practice. Information technology was regarded as a new competitive weapon to capture new economies of
scale by utilizing machinery, space energy and specialized labor efficiently (Parsons, 1983). CAD, database, e-mail were deployed in product design and development. CAM (Hatvany, 1984), quality assurance system (Nassar and Souder., 1989) and quality monitoring system was utilized at the operational stage. An Integrated system for manufacturing was advocated to integrate both CAD/CAM system and to streamline the whole product development cycle from product design, manufacturing process simulation to final product fabrication (Allen, 1989; Encarnacao, 1984; Scott and Daily, 1988). Standardized databases, graphics and operation systems accelerated the product development process and eased data transfer between systems. IGES format emerged as a standard for data exchange between CAD/CAM systems helping to solve database communication problems (Pavlakos et al., 1989; Smith and Wellington, 1984).

A knowledge based geometric modeling system was developed for preliminary design using an object-oriented approach, and Object-Oriented Programming (OOP) was comprehensively used for flexible design (Akagi and Fujita, 1989). Concerning the order quantity and production capacity, rational decisions about production quantity and sub-contract orders were made. Object-oriented representation was able to help to decompose a system into primitive functional levels. Maintainability as well as core-sharing via inheritance and data encapsulation was a critical feature of object-oriented systems (Arnold and Early, 1989; Kraemer, 1989; Lockemann, 1989).

In 90's, there was an increasing need to integrate artificial intelligence harmoniously with design practitioners and maximize profit from cultivating advanced design, automation, and production. Key issues on the development of expert systems for integrated product and process development included system
architecture (Chen and Power, 1990), knowledge representation, knowledge based
design and maintenance, problem solving paradigms and knowledge sharing (Zha et
al., 1999; Sieger and Salmi, 1997). Expert systems incorporate advanced object
oriented modeling for product and process modeling (Gorti et al., 1997; Erens and
McKay, 1994), constraint-based language, rule-based reasoning, case based reasoning,
genetic algorithms and fuzzy logic, and have an interface with which to analyze and
evaluate programs, databases and CAD systems. The comprehensive use of the
Internet and web applications greatly enhances the communication and collaboration
between geographically dispersed manufacturers, perhaps culminating in the creation
of a "virtual enterprise" which is a temporary, culturally diverse geographically
dispersed, electronically communicating work group (Jarvenpaa et al., 1998). Market
surveys can be distributed to the potential users. The marketing department of the firm
can list out data based on past purchase records through e-mail and web-sites. Media
products simulate the actual functions and appearance of the physical product and the
Internet provides a test-ground for new prototypes. Feedback can be obtained through
potential customers' behaviors and messages (Muller et al., 1996). CAD systems
associated with expert system optimize design parameters and improve product design
in regards to its cost (Howarth, 1994). One or more than one programming model
includes linear programming (Chandra and Shall, 1988), Integer programming
(Liberatore, 1984; Ghosh and Wysk, 1987), Markov model (Monahan and Smunt,
1989), Stochastic programming (Fine and Freund, 1990), dynamic programming and
goal programming (Ignizio, 1976) are used to resolve the problems of Computer
Integrated Manufacturing (CIM). After sales support and service is becoming more
important in the product life cycle. Customer Relationship Management (CRM) is
advocated to strengthen customer loyalty and enhance brand differentiation.
In 21st century, numerous researches have been conducted on collaborative application portal and make use of web technology to achieve effective and efficient communication and collaboration of various parties in product development activities (Zhan et al., 2003; Lan et al., 2004; Huang, 2002). Knowledge based system is applied in product development and it helps to leverage knowledge representation, usage, and interpretation to help reengineer the product development life cycle (Zha et al., 2001; Shehab, 2001).

2.1.2 Research on Object Technology, Case Based Reasoning and XML

A number of research issues must be addressed in order to achieve responsive product development. During design and analysis, modeling is an inevitable step. The model can represent a complex reality with all the features of that reality necessary for carrying out the current task in a simple way (Brown, 2002). Object models help to analyze the businesses scenario and provide abstractions with which programmers can develop OOP. UML is widely used for system modeling since it has competitive advantages on modeling application structure, architecture and product development process when comparing with the most common business modeling technique (the tree hierarchy mode). The simple core notation offered by object modeling helps system analysts to gain a better understanding on the business process and application behaviors. With object modeling techniques, the sophisticated activities can be identified and expressed, clearly, without ambiguity. Since product development processes are dynamic and coherent, object technology is proven to possess the capability of representing the co-relation of each agent, supplier and customer effectively. The features of inheritance and polymorphism also help to maintain the data structure of information system more readily in the turbulent market. Data play a
crucial role in system applications. The new emergence of XML is attracting not only academic attention but also commercial software providers' attention since XML enables data interchange between heterogeneous systems and it can be deployed together with HyperText Markup Language (HTML). Data has to be analyzed and computed to provide useful information to users and numerous systems have been designed to exploit domain specific knowledge for problem solving.

2.1.2.1 Product Data Management (PDM) System Modeling with OO Approach

Manufacturing firms today face the challenge of a changing environment such that a good PDM system is required to manage data efficiently in terms of business workflow and production process. PDM is designed to optimize the engineering process by coordinating and controlling access to a vast amount of information (Kovacs, 1998), providing a document management service which can be used by engineering software applications to access, control and manipulate the data within complex manufacturing lifecycles.

The traditional procedural approach concerns the contents of processes and the steps; while the object-oriented approach contributes to a better cognitive model (Singer, 1996). Object technology, which is a modeling technique to interpret the real world with entities as objects with distinct identities and characteristics, has been applied in PDM. Object technology guides relevant practitioners to think about the problems of entities and their relationship with customers (Brown, 2002). The impact of the object technology is that it changes the mindset of the programmers and has its own terminology such as object, class, behavior, property, inheritance and polymorphism. Object Technology proved its usefulness in terms of reusability and reconfigurability in various business environments in the last decade (Cohen & Booch,
1996; Fayad et al., 1996; Fedorowicz & Villeneuve, 1999). In general, object technology is used in areas related to computer aided process planning (Law & Tam, 2000; Luo, et al, 1997) and assembly systems (Maione & Piscitelli, 1999; Ho & Ranky, 1997). However, further studies in other areas such as product development need more in-depth investigation to capitalize on the merits of this emerging technology.

2.1.2.2 Knowledge Based System for Product Development

In recent years, there has been an increasing interest in AI research that is concerned with the deployment of Case Based Reasoning (CBR), fuzzy logic, Artificial Neural Network (ANN) and Genetic Algorithm (GA) in various decision support systems (Dhar 1997). Early product development is based on incomplete, ill-structured and poor quality information (Haque et al., 2000) which implies that the nature of the experience is fragile and not well-organized. As a result, knowledge of new product development is difficulty to be represented as fuzzy rules in fuzzy system which is used for approximating a function. Although fuzzy logic can handle uncertain, vague and imprecise information, it has limited abilities to handle missing or incomplete data that leads to fuzzy logic unable to make reliable prediction (Cox, 1995). ANN generates rules by learning from the training examples without human interference. However, the training process is quite time consuming and it is quite difficulty to correct the training pattern so ANN is not the best solution for iterative product development. GA is based on natural evolution processes in which a series of genes or chromosome are the basic organism. Each piece of information for the problem is encoded in the chromosome as a sequence of numbers. GA, which is one of the heuristic search and optimization techniques, imitates the natural selection and
evolutionary processes. Optimization is the process of reducing the space to one or a few of the best ones by setting the objective function and constrains. In general, the typical optimization techniques such as quadratic programming or multiple-integer linear programming are capable to search over the feasible domain of the variable in an effort to determine good values for the variables. However, once there are many variables or constrains and the relationship between various variable is unclear, the solution is difficult to be found and uncertainty may undermine the confidence or accuracy of the proposed solution.

CBR is one of the AI techniques applied in product design for the diagnosis of engineering problems. The idea of CBR is to utilize the specific knowledge of a particular situation (case) and solve a new problem by matching it with the previous cases stored in a case repository (case base). The new engineering problems are solved when the past similar cases are retrieved for adaptation. The ability of the system to learn also enables the CBR system to possess human-oriented innovation to assist design practitioners for new product design. As a result, CBR is the preferred AI technique used for product development as CRB can cope with unknown factors which continuously permeate in the design process.

Results based on the investigation into case-based design processes have led to the creation of several case based tools which advise designers on various alternative designs. CBR and rule based systems are applied to suggest initial values and lay-out choices for aircraft design based on similarities with the problem specifications (Rentema et al., 1998). When functional and/or geometrical constraints have been violated, the parameter values have to be modified. General suggestions for the modifications are formalized by rules. CADET (Sycara et al., 1992) provides
conceptual design aid for electro-mechanical devices. An Index transformation technique facilitates the combining of multiple cases to resolve novel problem.

The first commercial CBR application, CLAVIER was developed by Lockheed Missiles and Space Company in Sunnyvale California (Hinkle & Toomey, 1994). The problem of Lockheed is that composite material, which consists of layers of carbon-fiber products, is formed into single laminated components by curing in an oven called an autoclave. CLAVIER stores each layout case in terms of parts, relative position of other parts and production statistics such as start and finish time, pressure and temperature. Case adaptation is done by substituting a mismatching part with a similar part in a similar place. Table 2.1 shows the featured project of applying CBR to product design. This includes CADET (Navinendra et al., 1991), CLAVIER (Watson, 1997), COMPOSER (Purvis & Pu, 1998), and CBS-TX (Kwong & Tam, 2002).

Table 2.1 CBR project related to product design

<table>
<thead>
<tr>
<th>Application area</th>
<th>Featured Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>CADET</td>
<td>CLAVIER</td>
</tr>
<tr>
<td>Conceptual design of hydro-mechanical systems</td>
<td>Layout of the components in autoclave</td>
</tr>
<tr>
<td>Behavioral specification of mechanical system and behavioral characteristics of mechanical components.</td>
<td>Name of parts, Table on which parts place, Parts location, Relative position of other parts, Production statistics</td>
</tr>
<tr>
<td>Case retrieval</td>
<td>Adaptive search algorithm and index transformation technique</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>Case adaptation</td>
<td>Algebraic design constraints</td>
</tr>
<tr>
<td>Case maintenance</td>
<td>---</td>
</tr>
</tbody>
</table>

One of the information system proficiency characteristics for agile manufacturing is migration, where IT/IS infrastructure paves the way for future electronic interactions with customer and suppliers (Dove, 1996). Information exchange between suppliers and customers is regarded as horizontal migration while information exchange between distributed repositories for completion of downstream processes is regarded as vertical migration. Data migration methodology is deployed when the application needs to retrieve the data from one database and reconfigure the structure of data to another knowledge base (Borgida and Brachma 1993; Chen et al. 1999).

Recently, the relationship between database and case base has increasingly received attention. Case base, which is a case repository storing the cases as knowledge representations, requires updated data in order to enhance the reliability of
the case. Kamp et al. (1998) discussed the most prominent current uses of database techniques within CBR. One is using a database management system as a persistent case store and the other approach is to realize a CBR system with database means. CLAVIER, which is the best example of the first approach, builds a case library based on a small database. The second approach is deployed by Li and Yang (2000) who proposed an integrated system that combines a CBR system with an active database system. It is suggested that the syntax "trigger" (a special kind of stored procedure of Microsoft® SQL Server™) is used to perform the rule mechanism which is executed automatically when the specified data modification occurs. Another issue is how to make the CBR system handle incomplete and vague data. Similarity measure (Motro, 1988), probabilistic datalog (Fuhr, 1993) and Dempster-Shafer Theory (Lee, 1992) have been proposed to tackle this problem but the missing information gap between the database and the case base has not yet resolved.

In practice, the source of the case knowledge is normally associated with the corporate database system, which, in most cases, is the corporate Relational Database Management System (RDBMS) (Irgens, 1995) such as Microsoft® SQL Server™. However, due to the fundamentally different data formats of CBR and the relational database, there is always the problem of data interchange between these two repositories (Chung and Lau, 2000). Kate, which is commercial CBR software, makes use of Data-parallel Bit-serial C (DBC) and Object Database Connectivity (ODBC). A Data-parallel Bit-serial C file, which is a script file for data migration, contains the following parameters: driver's name; database name in the ODBC data source, column name and table name containing the identifiers of the cases, the status of the case and column name as well as the table name that contain the slots of the cases. (Kaidara, 2000). There are various migration methods processed by means of ODBC
(Couper et al. 1999; Guan et al. 1998), XML (Burkett 2001; Singh 2001; Sundaram 2001) federation approach (Hohenstein, 2000; Hamilton and Chilton, 1995) and object exchange model (Papakonstantinou et al., 1995).

2.1.2.3 Existing Product Data Exchange Standard

(I) Product Data Markup Language (PDML)

PDML is an XML vocabulary designed for integrating and interoperating business processes across US Department of Defense and contracting organizations which deploy the PDM system for managing the product design and production data within the supply network (Burkett, 2001). PDML is being developed as part of the Product Data Interoperability (PDI) project under the sponsorship of the Joint Electronic Commerce Program Office (JECPO). It is supported by several other Federal Government agencies and commercial entities. PDML adopts STEP (ISO 10303) as the data specification language that controls the semantics of data exchange with XML encoding and the Internet as an integrated platform for product data exchange. The vision of PDML is to enable users to obtain accurate and authoritative data in a timely manner, and easily without resorting to thesauri or indices. PDML is formulated with three core components which are Application Transaction Sets (ATS), Integration Schema and Mapping Specifications. ATS is an XML Data Type Definition (DTD) that specifies the data elements and structure for users from different communities. Seven ATS are defined in PDML which are product structure, product description document, technical order-4, JEDMICS, MILD-STD-2549 DIP1, MILD-STD-2549 DIP3, and MILD-STD-2549 DIP7. Since the data needs to be integrated from various parties to provide an intermediary view as Integration Schema, mapping specification specifies the relationship between
Integration Schema and ATS and applies the conversion rules to convert the ATS to a neural integrated representation.

(II) PDM Enabler

The PDM Enabler, which defines an interface enabling the interoperability of a PDM system, was developed by the Manufacturing Domain Task Force of Object Management Group (OMG). OMG is the software consortium that promotes the theory and practice of Object Technology (OT) for the development of distributed computing systems. OMG would like to make a proposal to establish standard interfaces and object framework for the enabling service provided by the PDM system. According to the joint proposal to the OMG in 1998, PDM Enabler is designed for allowing compliant interface to be built both on PDM systems with an Object Management Architecture (OMA) infrastructure on the existing non-OMA legacy PDM system. PDM Enabler also provides a framework for a PDM system interfaces that can be readily customized and extended by software suppliers and customers. The design characteristic of PDM Enabler is that the primary PDM constructs are the first-class Common Object Request Broker Architecture (COBRA) objects and apply the approved CORBA specification including Life Cycle Service, Object Property Service, Relationship Service, Time Service and Currency Service. The specification provides services or facilities in the PDM domain rather a complete set of interfaces for implementing a fully functional PDM system. The PDM Enabler includes twelve Interface Definition Language (IDL) modules that are correlated to eight PDM enablers. These are:

i. Request for Engineering Action

ii. Engineering Change Order

iii. Manufacturing Implementation or Release to Manufacturing
iv. Document Management for the Enterprise
v. Product Structure Definition
vi. Effectively of Products and Occurrences
vii. Configuration Management
viii. Test, Maintenance and Diagnostic Information

The enablers define the necessary interface and depict the PDM processes within the proposed object framework. *Enabler is a term derived from Total Quality Management (TQM) principles. An enabler is defined as a physical entity (a machine, human activity or computer program) that implements or supports an abstract business process* (OMG, 1996). The ultimate goal of enablers is to empower product development groups by providing agile means to access and manipulate data among different PDM systems.

(III) PDM Schema

The PDM Schema (STEP PDM Schema, 2002) is a reference information model for interchanging a central, common subset of the data being managed within a PDM system. PDM Schema is the joint effort of ProSTEP and PDES Inc. It represents the intersection of requirements and data structures from a range of STEP Application Protocols (AP 203, 212, 214, 232) which are in the domains of design and development of discrete electro/mechanical parts and assemblies. Similar to the PDM Enabler, the STEP PDM Schema is not a specification for the functionality required for the complete scope of all PDM system functionality – i.e., it is the intersection of functionality present in the set of STEP Application Protocols. The STEP PDM Schema is a core set of entities in STEP that support the mapping of concepts for PDM. The objective of establishing the PDM Schema is to promote interoperability
between STEP application protocols in the area of product data management. The modular semantic units of functionality are listed below:

i. Part Identification,

ii. Part Classification,

iii. Part Properties,

iv. Part Structure and Relationships,

v. Document Identification,

vi. Document Classification,

vii. External Files,

viii. Relationships between Documents and Constituent Files

ix. Document and File Properties,

x. Document and File Association to Product Data,

xi. Document and File Relationships,

xii. Alias Identification,

xiii. Authorization,

xiv. Configuration and Effectively Information,

xv. Work Management Data.

(IV) Product Data eXchange (PDX)

PDX is a standard developed by IPC which is a trade association that interconnects all of the players in this industry: designers, board manufacturers, assembly companies, suppliers, and Original Equipment Manufacturers (OEM) (Cover Pages, 2001). As the problem of communicating product content information between OEM and Electronic Manufacturing Service (EMS) providers and component suppliers arises, PDX is designed for providing an effective way to exchange product content and changes in a common language and this helps to

2.17
enhance the efficiency throughout the supply chain since they are all partners. The leading corporations and standards organizations in electronics manufacturing includes NIST, Intel, Agile, Digital Marker, Marshall, IPC, PeopleSoft, Solectron, Unicam, etc. form the standard group which is a functional group of the National Electronics Manufacturing Initiative, Virtual Factory Information Interchange Project (NEMI VFIIP). The PDX standard is based on XML to encode structured data into a format which can be read by both humans and machines. IPC 2571 states: "The Product Data eXchange 1.0 standard defines an XML encoding scheme that enables a total product definition to be described at a level appropriate to facilitate supply chain interactions." The scheme is defined for Bill Of Materials (BOM), Approved Manufacturer List (AML), changes and references to documents describing geometric and other definition characteristics. Table 2.2 shows the features and functions of various data exchange standard.
<table>
<thead>
<tr>
<th>Functional Objective</th>
<th>PDML</th>
<th>PDM Enablers</th>
<th>STEP's PDM Schema</th>
<th>PDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation Objective</td>
<td>Integration</td>
<td>Access</td>
<td>Interfacing</td>
<td>Supply Chain interaction</td>
</tr>
<tr>
<td>Implementation Objective</td>
<td>Accommodate: 2549, STEP APs, XML, PDM Enablers, PDM Schema</td>
<td>PDM vendor</td>
<td>AP 203/214</td>
<td>IPC 2570, 2540, XML</td>
</tr>
<tr>
<td>Customer Requirement</td>
<td>Government</td>
<td>PDM Vendor/user</td>
<td>PDES, Inc. &amp; ProSTEP</td>
<td>Electronic Manufacturing Service Provider</td>
</tr>
<tr>
<td>Scope</td>
<td>Product Data</td>
<td>PDM System Only</td>
<td>Subset of AP203 &amp; AP214</td>
<td>IPC 2570</td>
</tr>
<tr>
<td>Schema</td>
<td>Based on STEP IR</td>
<td>Object Model (UML) base on STEP</td>
<td>STEP APs</td>
<td>DTD based on IPC2570</td>
</tr>
<tr>
<td>Transaction Sets</td>
<td>XML</td>
<td>IDL</td>
<td>Express/P21</td>
<td>XML</td>
</tr>
<tr>
<td>Data Sharing Method</td>
<td>XML/File exchange</td>
<td>CORBA/Access</td>
<td>P21/SDAI</td>
<td>Product data exchange packet</td>
</tr>
<tr>
<td>Validation</td>
<td>During POC- use real world data/environment</td>
<td>Vendor product based</td>
<td>PDES Inc/ProSTEP Roundtable(pay to play)</td>
<td>DTD</td>
</tr>
<tr>
<td>Access Method</td>
<td>PDM Enablers, SDAI, XML SGML, ODBC</td>
<td>CORBA</td>
<td>SDAI</td>
<td>XML SGML</td>
</tr>
<tr>
<td>Proposed by</td>
<td>US federal government (Department of Defense) Several Major PDM Vendor</td>
<td>PDMIC</td>
<td>Companies in consortium includes Boeing, British Aerospace, IBM, NASA, NIST, Rockwell, etc</td>
<td>IPC (trade association)</td>
</tr>
</tbody>
</table>
2.1.2.4 Review of XML Adoption in Industry

Data communication among heterogeneous systems is always a concern particularly in situations where the support of a universal data exchange standard is unavailable. XML is a new emerging technology which can manipulate, access and exchange data or metadata over different platforms and systems. XML is regarded as next generation data representation (Manola, 1999). Burkett proposed Product Data Markup Language (PDMML) (Burkett, 2001) which offers a new paradigm for product data exchange based on existing technologies that facilitate the integration and interoperability of the business processes of the Department of Defense and a contracting organization. XML is also recognized as a data structuring or encoding scheme with specified data structure design principles with Document Type Definition (DTD) (Ceponkus, 1999) design. Manufacturability Markup Language (MML) (Shiau et al., 2000) also incorporates XML to exchange manufacturability requests and product information between different agents, providing an overview of key domain models and their integration within a decision-making environment. It combines the distributed object standards with the web standards and protocol, creating an object web in an Integrated Product and Process Development (IPPD) (Rezayat\(^1\), 2000) environment. This approach allows individual access to the enterprise knowledge, providing decision-support functions to achieve effective design and manufacturing. Rezayat\(^2\) (2000) has further studied XML for defining the request interface to achieve effective knowledge reuse where one IP3D strategy is referred as Knowledge Based Product Development (KBPD). Whilst XML is applied to model object-oriented framework, a retrieval technique for structured documents is utilized to support effective retrieval of reusable object oriented modules (Lee, et al,
XML, which acts as a web standard and protocol to manage metadata, aggregates with PDM to form a more complete solution for enterprises.

XML, which is applied in manufacturing to facilitate efficient data exchange, is the web-based technology integrating various systems and applied in several areas such as supply chain management (Turowski, 2002; Chen, et al., 2001; Fung et al., 2001), PDM (Rezayat, 2000; Burkett, 2001) and shop floor control. XML acts as a document media in the supply chain where XML helps assemble information from a variety of sources and encodes information in a common intelligent electronic format (Chen et al., 2001). Deploying XML in agent-based e-commerce helps automate the inter-company business processes and communications (Turowski, 2002). Several tools such as SISCO, SIMPLEX have been developed to facilitate the smooth operation of the supply chain. SISCO (Chatfield & Harrison, 2001) is a java-based tool developed for simplifying supply chain simulation model development whereas SIMPLEX (Buxmann et al., 2001) is a java-based prototype for displaying, describing, exchanging, converting and integrating XML documents between partners in a supply chain network. Collaborative-commerce (C-commerce) is proposed (Rayson, 2001) in order to allow customers and diverse members of the supply chain to collaboratively design, build, market and deploy products and services by means of web technology to access and manipulate critical information.

Apart from sharing information externally, Collaborative Product Commerce (CPC) focuses on internal communication within the department for developing novel and innovative solutions, tailored to product development and customer service (Rayson, 2001). XML is deployed to manage product data with the ability to handle a large amount of dynamic content. A product knowledge markup language based on key characteristics as a main enabler for defining the ontology is implemented as part
XML, CORBA and Java are used to build a distributed intelligence control architecture model for integrating a flexible manufacturing system. For the purpose of achieving distributed collaborative design and manufacturability, MML incorporating CORBA provides a rapid manufacturing feasibility assessment tool which can be used at different design and planning stages (Shiau et al., 2000). Based on the Principle of Increasing Precision, Decreasing Intelligence (IPDI), Kolluru et al. (2000) suggest an intelligent model that imparts 'perception-decision-action' to machines and equipment on the manufacturing shop floor where XML is used to communicate multi-model information. On the other hand, Microsoft's Windows Distributed Internet Applications for Manufacturing (DNA-M) architecture is used in plant operations where Component Object Model (COM) integration, Visual Basic for Applications (VBA) and XML are used to develop decision support tools (Arora, 2000) while Object Linking and Embedding (OLE) is employed for process controls.

2.2 An Exploratory Study in Adoption of Information Systems for Product Development

This section examines the extent to which information system for product development, as obtained from a broad literature review, are being used effectively in companies. The objective of the research is to find out the existing practice of the manufacturers during product development. It investigates the comprehensiveness of deploying information systems to assist knowledge workers during product development. The literature about the influence of information systems on product development lifecycle outcomes shows mixed results in the Pearl River Delta (PRD). The survey respondents are mainly from the electronics industry. Various issues (such
as existing practice of product development, system development for the product life cycle, product data exchange during product development and potential features of the information system) have been identified and assessed in Hong Kong (HK) and People’s Republic of China (PRC). The methodology of the survey is then presented in order to give an overall picture on how the entire survey was carried out from beginning to end. The findings are drawn and implications of this survey are discussed.

2.2.1 Related Studies on Information Systems for Managing Product Lifecycle

According to the survey (TDC Trade¹, 2003; TDC Trade², 2003) conducted by Hong Kong Trade Development Council (HKTDC), it is confirmed that HK is a crucial business platform for the United State and European companies with operations in the Pearl River Delta (PRD). Over 80% HK manufacturers have established factories in China. HK manufacturers transmit data by Digital Subscriber Line (xDSL), Local Multipoint Distribution Service (LMDS) and Fibre-to-the Building (FTTB), cable modem or other technologies. 54.5% of the establishments in HK use personal computers and 44.2% of the establishments have internet connections (Census & Statistics Department, 2003).

Information systems have been widely used for managing product lifecycles. The typical technology used is object technology which is generally used for object oriented design, analysis, modeling and programming. According to the current features of manufacturing, the model is dynamic and consolidated; has coherence and integrity. The object technology helps express this dynamic system by studying the relationship, behaviors and attributes of business/manufacturing objects in order to
respond to the changing business environment. The main objectives of applying object technology are to eliminate the semantic gaps between the data model and existing practice (Khoo et al., 2002), optimize the development process and realize data sharing. As a result, the design of manufacturability, cost-effective and tailor-made products can therefore be achieved rapidly and flexibly. Table 2.3 summarizes the recent researches involving the employment of object technology to develop manufacturing systems.

<table>
<thead>
<tr>
<th>Object oriented data design and analysis</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gorti, 1999</td>
</tr>
<tr>
<td></td>
<td>Khoo et al., 2002</td>
</tr>
<tr>
<td>Modeling</td>
<td>Christiansen and Vesterager, 1999</td>
</tr>
<tr>
<td></td>
<td>Liu et al., 2003</td>
</tr>
<tr>
<td></td>
<td>Xu and Wang, 2002</td>
</tr>
<tr>
<td>Programming</td>
<td>Gayretli and Abdalla, 1999</td>
</tr>
<tr>
<td></td>
<td>Shehab and Abdalla, 2001</td>
</tr>
<tr>
<td></td>
<td>Zha, 2000</td>
</tr>
<tr>
<td></td>
<td>Zha et al., 2001</td>
</tr>
</tbody>
</table>

Artificial intelligence is commonly adopted in sophisticated Knowledge Based System (KBS) where Case Based Reasoning (CBR), fuzzy logic, Artificial Neural Network (ANN) or qualitative reasoning are usually adopted to increase the value of KBS. The main idea of CBR is that analogies can be drawn between past situations, generally known as cases, and the current problem to be solved. Fuzzy logic and inference techniques are also applied to deal with uncertain data and knowledge in the design process. The neural network approach is well suited to this task as it includes cost estimation (Cavalieri et al., 2003; Zhang et al., 1996), process optimization (Lin et al., 1995; Wilcox and Wright, 1998), market analysis (Chen et al.,
2002; Ishihara et al., 1997) and deploys Quality Function Deployment (QFD) to
generate the desired output by considering customers’ requirements and priorities
(Myint, 2003).

2.2.2 Design of Questionnaire and Data Collection

The survey was distributed through mail or e-mail where the database was
provided by Department of Industrial and Systems Engineering of The Hong Kong
Polytechnic University. A database of 1470 manufacturing firms has been created
based on industry section defined by HKTDC. The database contains the name of
the company, location, contact person, his/her title and postal address. Another 55
questionnaires were distributed to MSc Students. Target respondents are
higher/intermediate managerial, administrative or professional persons working on
product design and development. The first questions are those which are relatively
easy to answer while the more difficult or threatening ones are left to the end. From
the 1525 questionnaires sent out 136 questionnaires were returned and 95 answered
questionnaires were suitable for analysis.

Three different types of questions, behavioral, attitudinal and classification
questions were set. The questionnaire contained five sections which are

i. Survey of Existing Practice of Product Development;

ii. System Development for Product Lifecycle Management;

iii. Product Data Exchange during Product Development;

iv. The Potential Feature of Information System and

v. Respondents’ Background Information.

Section “A” contains six questions probing the difficulties found in product
development and the resolution method. Section “B” explores the deployment of
Section "C" contains open-ended questions to let users illustrate how they solve the problem of data exchange, multiple choices to allow respondents to select data exchange methods and ranking questions to find out what is the most beneficial to a respondent. Section "D" was designed to collect opinions of respondents about the potential features of the information system. The given answers can motivate academic or commercial companies to have further development in PLM. A structured questionnaire has been developed on a five point Likert scale (1—strongly disagree, 5—strongly agree). The scales may ask the respondents whether or not they agree with the subject. Section E, which includes classification questions, provides a profile of the respondents by finding their position in the company, industry sector and the number of employees in the manufacturing or other sector.

In HK, Small and Medium Enterprises (SMEs) are defined as enterprises engaging less than 100 persons in the manufacturing sector and less than 50 persons in the other sectors. However, most companies moved their production plant to the PRD, as this allows enterprises to operate as labor intensive manufacturing firms and hire more labor at a low rate of pay in China. It was found that 41 out of 95 responding companies have more than one thousand workers in the manufacturing sector. HK acts as a regional headquarters and provides a service platform to support their PRD business. 42 out of the 95 responding companies hire more than 100 staff and 41 out of 95 responding companies employ less than 50 staff in HK in the provision of financial and business services, trade and related services, transport & logistics services and communication & media services.
2.2.3 Observation

Product development is full of challenges and manufacturers need to face various problems. Most customers approach the manufacturing firms with merely a design concept and unclear customer requirements that makes manufactures further clarify and define the specification for the products. Manufacturers, who produce sophisticated products, face the problem of insufficient production capacity, and technological constraints. Today’s manufactures need to find out the optimal solution by paying attention to the delivery date, technology and capacity constraints. Most respondents realized that unclear customer requirements (57%), short delivery time (56%) and technological issues (49%) are the most common problems found in the product development process. Design-Analysis-Redesign, which is commonly found in the initial stage of product development, leverages the inherent iterative nature of design. Regarding the brief customer requirements, some design solutions are proposed and evaluated by customers. If those solutions do not perfectly match with customers’ criteria, design practitioners are required to revise the draft or redesign the products. Table 2.4 shows that the same design change occurs repetitively in the development of other new products and the number of design changes that companies usually encountered in the product development process is usually less than 10.

Table 2.4 The co-relation between the no. of design changes and changes in the same design

<table>
<thead>
<tr>
<th>No. of changes</th>
<th>Same design change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-5</td>
</tr>
<tr>
<td>Yes</td>
<td>30</td>
</tr>
<tr>
<td>No</td>
<td>12</td>
</tr>
</tbody>
</table>
Although information systems and technology has developed rapidly, most of the respondents resolve the problems with their experience (76%) and most of them brainstorm with their team member (52%) or adopt supervisor’s advice (39%). It should be noted that companies do not own any intellectual capital. Only the knowledge workers in the company possess knowledge and intellectual capacity and the intellectual capital is dynamic and mobile since human experts may move from one company to another. Around 70% of the respondents realized that the loss of corporate knowledge is due to the resignation of experienced staff and 60% of them claimed that their companies have a methodology to manage the knowledge. Table 2.5 summarizes the technology-based approaches adopted by PRD manufacturers including Knowledge Based System (KBS), documentation, library of knowledge (database) and intranet as infrastructure of knowledge sharing.

Table 2.5 Knowledge Management (KM) methodology adopted by PRD manufacturers

<table>
<thead>
<tr>
<th>Method to KM</th>
<th>No of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>14</td>
</tr>
<tr>
<td>Database</td>
<td>9</td>
</tr>
<tr>
<td>KBS</td>
<td>16</td>
</tr>
<tr>
<td>Intranet</td>
<td>3</td>
</tr>
<tr>
<td>Meeting, training, teamwork</td>
<td>5</td>
</tr>
<tr>
<td>No specify</td>
<td>8</td>
</tr>
</tbody>
</table>

KBS encodes the expert knowledge stored in the knowledge repository. The inference engine usually incorporates artificial intelligence techniques that imitate human intelligence to resolve problems. Data analysis (65%) is ranked highest while some respondents use an information system for computer aided design (47%) decision support (38%) and problem solving (25%). PLM as a concept within
operation management is concerned with both the process of designing products and also with production systems, realization of production, distribution, and service. However, only 19% of the respondents have developed or bought the PLM system.

Among those having PLM system, it is found that 39% of the respondents adopted object-oriented design and analysis; 28% adopted object-oriented modeling and 28% utilized OOP to develop an information system. Although most academic institutions and IT consultancy firms have experience with expert systems using a variety of AI, few manufacturing firms in PRD have successfully developed and deployed AI application programs. It would be of interest to find which respondents have developed a hybrid system with more than one AI technique. The ranking of AI techniques used in PLM systems is shown below.

i. Rule based,

ii. Fuzzy Logic,

iii. Neural Network

iv. Case Based Reasoning

Domain-specific knowledge is stored in a knowledge base and the functionality of KBS will be dramatically reduced if knowledge is incomplete. To enable the proper use of the captured knowledge, it is important that available data can be organized as information which is then systemized to form knowledge, thus allowing the reuse of expert advice in an efficient way. About half of the respondents found difficulties in data exchange. Format difference is the major barrier to data exchange. Another common data interchange problem is lack of software to open customer or vendor files. Table 2.6 summarizes the problems encountered by the respondents.
Table 2.6 The problems encountered by the respondents

<table>
<thead>
<tr>
<th>Problems related to data exchange</th>
<th>Number of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format difference</td>
<td>15</td>
</tr>
<tr>
<td>Data loss</td>
<td>1</td>
</tr>
<tr>
<td>Software version</td>
<td>1</td>
</tr>
<tr>
<td>Security problem</td>
<td>2</td>
</tr>
<tr>
<td>Proprietary system</td>
<td>2</td>
</tr>
<tr>
<td>No required software</td>
<td>4</td>
</tr>
</tbody>
</table>

The emergence of PDF and XML helps solve those problems as Acrobat Reader® and Internet Explorer are free; Microsoft Office is sold at a reasonable price and is commonly used by companies. In particular, the CAD standard—STEP/ISO10303 provides a methodology for representing and exchanging digital product information. Table 2.7 shows how the respondents resolve the problems of data exchange with business partners.

Table 2.7 Solutions for data exchange difficulties

<table>
<thead>
<tr>
<th>Problem Resolution Method</th>
<th>Number of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting</td>
<td>12</td>
</tr>
<tr>
<td>Transfer to CAD standard</td>
<td>11</td>
</tr>
<tr>
<td>Handle by technician</td>
<td>6</td>
</tr>
<tr>
<td>Downgrade the data</td>
<td>1</td>
</tr>
<tr>
<td>Use traditional data format (Fax, phone, hardcopy)</td>
<td>10</td>
</tr>
<tr>
<td>Install software</td>
<td>10</td>
</tr>
</tbody>
</table>

Despite the emergence of advanced data exchange techniques, some firms may still use the traditional approach, that is, printing it as hardcopy when they cannot solve the problem of electronic data exchange. Although the problem seems to be solved instantly, a long-term solution should be implemented to minimize duplication of data entry, reduce typing errors, shorten data exchange time and maintain data
integrity. The benefits of the electronic data exchange method is measured on a 4-point scale (4—most beneficial, 3—Average beneficial, 2—somewhat beneficial, 1—not beneficial) and the results are summarized in the Table 2.8.

Table 2.8 Mean scores, rank and standard deviation of the benefits of data exchange

<table>
<thead>
<tr>
<th></th>
<th>Reduces typing error</th>
<th>Minimizes duplication</th>
<th>Shortens data exchange time</th>
<th>Maintains Data integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>2.36</td>
<td>2.87</td>
<td>3.13</td>
<td>2.96</td>
</tr>
<tr>
<td>Rank</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>1.12</td>
<td>0.97</td>
<td>0.92</td>
<td>1.00</td>
</tr>
</tbody>
</table>

In order to explore the potential features of the future information system, three distinguishing features, which are adaptability, configurability and automatically updating of information, have been set. Respondents consider the information system with configurable methods to cope with the dynamic market as the most important and the function of "automatic update" is ranked as second. Adaptation, that looks for prominent differences between the past case and current case such that the information system adjusts the stored solution to fit the needs of the current case, is ranked as third. Table 2.9 lists the importance of the potential features for the future information system.

Table 2.9 Mean scores, rank and standard deviation of the of the potential features

<table>
<thead>
<tr>
<th></th>
<th>Adaptive</th>
<th>Configurable</th>
<th>Automatic update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>3.49</td>
<td>3.66</td>
<td>3.66</td>
</tr>
<tr>
<td>Rank</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.85</td>
<td>0.87</td>
<td>1.15</td>
</tr>
</tbody>
</table>
2.2.4 Discussion

A response rate of 8.9% is predicted. Updated information of participants (address and contact person) helps increase the response rate. Since most respondents come from the electrical and electronic sector, benchmarking of various industrial sectors could not be performed. Second, a mono-respondent approach is adopted due to high cost associated with the multi-respondent approach. Third, due to the fact that only a small proportion of respondent have adopted object technology and artificial intelligence for the PLM system, the significance of the findings of section B (system development for product lifecycle management) is decreased. Despite the limitations mentioned, a pilot study was conducted on the technology based approach to knowledge management in PRD manufacturing firms.

Information technology acts as a facilitator for enhancing a firm's dynamic capability through knowledge management. An information system helps to manage repetition of design change. Change management has received a lot of attention recently and engineers or product designers should regard design change or engineering change as a way of continuous improvement of the product or process rather than a fire-fighting issue. Some repetitive design changes can be formulated as cases such that junior design practitioners can study and learn from the past case to help them to solve problems or prevent the similar problems from occurring again.

Although CBR is not ranked high among AI for PLM systems in this survey, most researchers realize that CBR as an effective knowledge management tool and the ease of capturing, encoding and eliciting knowledge become major advantages for SMEs and corporations.

A configurable information system helps increase the agility and responsiveness of a company to deal with a dynamic market. Adaptive systems
incorporating artificial intelligence assist junior staff to make decisions and resolve problems. The lower rank given to the self-learning feature of the information system reflects the fact that respondents wonder whether the suggestions or advice given by the information system is reliable. The result shows that the respondents, who are aware of data exchange difficulties, also realize the importance of automatic data exchange.

This survey reveals that most manufacturing firms face various problems including acquiring customer requirement and short delivery time. The root cause of the problem is the difficulties found in disseminating and exchanging information. The mature object technology and advance artificial intelligence techniques are proposed to be adopted in developing a responsive product development system. The further development of information system should include agility (configuration), self-learning (adaptation) and automatic data exchange with customizability in order to cope with dynamic market.

2.3 Implications from the Literature Review

Whilst there are many publications about applying object technology in manufacturing systems (Maione & Piscitelli, 1999; Ho & Ranky, 1997; Law & Tam, 2000; Luo, et al, 1997) there is a lack of literature dealing with interconnecting the object technology and XML in the product development process. In order to achieve knowledge integration to support product development, a Responsive Product Development System (RPDS), which enables dynamic data-knowledge conversion, is proposed in this thesis. There are some research activities conducted on data exchange with different information sources and on the maintenance of case based reasoning systems (Mileman et al., 2000; Shiu et al., 2000; Ferrario and Smyth, 2000) which
have led to an awareness of the importance of the updating of knowledge, but there are still few studies on the automatic transformation of raw data to an upper knowledge level stored in a case base. A HKM is proposed as it is characterized by its ability to integrate the heterogeneous data repositories, associated with CBR and relational databases to achieve efficient bi-directional data interchange. The literature review so far indicates that whilst there are a number of publications related to the approaches and methodologies to achieve efficient product data interchange, the research concerned with the deployment of universal data exchange to leverage knowledge in product development has not received the attention it deserves. The proposed PIML, which is a modified version of XML, is capable of providing a precise description of semantics by using object technology, and forms a universal data exchange standard.
CHAPTER 3. DESIGN OF A RESPONSIVE PRODUCT

DEVELOPMENT SYSTEM (RPDS)

This chapter focuses on the development of a Responsive Product Development System (RPDS) to support the process in the early stages of product development, featuring the capability to respond to customer changes in a timely manner. RPDS is proposed as a dynamic data exchange schema based on the integration of a case repository with a corporate database by virtue of object technology and the emerging XML data exchange standard, so as to attain the goal of responsive product development. In general, the distinct features of RPDS include (1) Agility for handling unanticipated change occurring in the early stage of the product development cycle (2) Self-learning for transforming data between heterogeneous systems (3) Customizability for fundamentally constructing the data schema to cope with an evolving product development process. In the design of the structure of RPDS, object technology, which is a paradigm shift of modeling and process activities, plays an instrumental role in accomplishing the essential functions of this proposed standard. Apart from object technology, other emerging techniques are also incorporated to form the backbone of this standard including (i) CBR for capturing past experience and knowledge, (ii) XML for exchanging information regardless of platforms or operating systems.

A cross-platform data exchange approach is proposed which is able to facilitate the creation of an environment conducive to managing engineering change and enhancing the activities of product development with object technology. One of the critical features of object technology, polymorphism, which helps to invoke various behaviors for the same message, overrides the pre-defined inherited operation
such that a flexible correlation can be formulated in the iterative product design stage.

In particular, RPDS provides a universal product information exchange standard through the Product Information Exchange Module which enables the knowledge workers to acquire sufficient and accurate information. The suggested approach achieves seamless data interchange based on neutral data files which serve to transform the corporate database into a case repository. PIML, which contains a specific set of XML for product development, takes advantage of the well-structured XML schema to provide a well-understood syntax and self-defined markup language to suit particular needs in terms of product data exchange.

This chapter provides an overview of RPDS which plays an instrumental role to support rapid product data integration and cope with unpredictable changes driven by internal and external customers. Section 3.1 pinpoints the feature of RPDS; Section 3.2 illustrates the underlying technologies for RPDS; Section 3.3 shows the system architecture of RPDS; Section 3.4 pinpoints the design consideration of RPDS. The final section 3.5 describes the details of each module of RPDS.

3.1 Features of RPDS

The features of RPDS are agility, self-learning and customizability which can be performed by Object Based Product Development Modules (OBPDM), the Hybrid Knowledge Module (HKM), and the Product Information Exchange Module (PIEM) respectively.

3.1.1 Agility

In the past decade, a paradigm shift in manufacturing has led to the need for it to be flexible, fast-responsive and agile. To survive in the increasingly customer-oriented marketplace, agile manufacturers should leverage up-to-date
information and co-operate with parties to master change and uncertainty, thereby enhancing competitiveness. Nowadays, design practitioners need to create products with innovative designs to meet customer demand in such a way that the company can gain a greater share in the market. Agile manufacturing requires an intelligent engineering design support system that can provide rapid evaluation of engineering designs and design changes. Object-oriented technology is used to organize, manipulate and process the information according to the real world objects that the information describes. Objects are basically "self-contained" units that encapsulate the information and algorithms to perform their works. The introduction of inheritance can reduce the complexity of product structure and define common operations in a superclass. As changes may be triggered by new customer requirements, modification of the process is required. Polymorphism, which can be instantiated and perform individual operations under specific subclasses, provides agility to help enterprises cope with unexpected events. This is especially suitable for turbulent market or manufacturing environments. OBPDMS applies the distinct feature of object technology, polymorphism, for handling engineering change in order to achieve the goal of responsive product development.

3.1.2 Self-learning Ability

Learning from past experience leads enterprises to enhance responsiveness to deal with a current situation that is similar to a past issue. The marketplace is constantly-changing. In order to adapt to those changes, a system is required to access and manipulate the data such that the knowledge representation can be supported by the up-to-date data. A corporate database is normally the mainstream repository used in companies to store recent data records while the knowledge base contains the facts
that describe the problem areas and the past experience, which are incorporated with a simulation of human reasoning. When the market substantially changes, the methodology of solving the problem may remain the same but the new set of parameters is adjusted to solve the new problem. This is called "parameter adjustment" which is a kind of structural adaptation. HKM makes it possible to exchange information across the traditional database and knowledge base by providing a seamless infrastructure for daily data entry into a relational database and facilitates knowledge acquisition with a knowledge toolkit. The proposed system provides an intuitive mechanism to replicate data in a heterogeneous system and promotes data integrity by eliminating manual data entry for data already available in the system. Another adaptation method in CBR is deviational adaptation. This is concerned with how to reuse the rules or formulas to generate a new solution from an old solution in the new situation. It requires the system to have an adaptive function. This is achieved by adaptive object-oriented software which supports evolutionary development and is especially suitable for use in the iterative product development cycle. Originally, adaptive object-oriented software was the nature of evolutionary software development. Since it is known that software reflects the real object world and the real product development situation is full of change, adaptiveness leads to the desired reduction of change impact. Adaptive object-oriented software, which has evolved from proven techniques for object-oriented software development, streamlines the actual product development process.
3.1.3 Customizability

Product design is regarded as an iterative process as lots of unknown factors continually permeate the design process. PDS is designed to be customizable in order to suit the knowledge workers' specific design processes and specifications. PIML is able to serve as a product information standard to manipulate the product data by means of a DTD/Schema. Apart from representing XML with XML Schema in static and structure format, another feature of PIEM illustrates how the dynamic behavior of the elements performs in order to attain the goal of acquiring and mastering the latest information. Apart from customizing the information for individual use, adaptation can be achieved by identifying the prominent difference between the current case and past case to resolve the new problem with formulas and rules. Generalization is used for finding out the common features of the cases which are more generally applicable to solving problems. Customization is deployed in the unique class/object in product development to cope with individual situations such that each class has its own behavior to react under various conditions. The abstraction operation is that one has no program code in the class but allows specific operations to be implemented by the subclasses.

3.2 Underlying Technologies for RPDS

RPDS deploys three major technologies which are: object technology, CBR and XML. Object technology capitalizes on the distinct features including inheritance, encapsulation, and polymorphism. Polymorphism is capable of invoking various kinds of object that behave differently for the same message. An object-oriented
method organizes information and processes in accordance with the properties and behaviour of real world objects.

In the rapid product development system, artificial intelligence is usually deployed and acts as a reasoning mechanism for emulating human's heruistic knowledge. In this research, CBR is deployed because cases store past experience and junior designers can be inspired by referring to past successful designs. A Case repository needs constant updating with new information from the corporate database to make solutions applicable to current problems. To enable data communication between the corporate database and the case base, HKM includes self-defined data schema that let users define a data hierarchy, for product data and export this in a neutral format for performing the data exchange.

Since design is an iterative process, there is no systematic product development flow and most data is loosely structured or of poor quality. PIEM is therefore developed as a universal product data standard. It structures the product data hierarchy using the Internet as an integration platform and XML as the data structuring and encoding language.

In general, a system is suggested incorporating the three technologies object technology, data exchange between relational database and CBR. It also uses a universal data exchange standard specified for product data to enhance the agility, self-learning and customerizable characteristics for achieving responsive product development. This combination is able to capitalize on the strengths of individual recomended practices and at the same time makes up an integrated system for product development. By utilizing object technology and XML, a novel product information structure was substantially formulated.
3.3 System Architecture of RPDS

RPDS is developed for the provision of a cross-platform data exchange approach which is able to facilitate the creation of an environment which can enhance the efficiency of the product development processes. The proposed system aims to leverage knowledge and novel technology and incorporate it into the product development process and resolve design and engineering problems by means of the inference engine. A rigorous and systematic approach is needed to attain the goal of being responsive to the market needs and changing customer requirements. This responsive product development has three aspects:

i. Data is the fundamental source of useful information. Object-oriented modeling is regarded as a generic modeling technique for system application which helps to capture knowledge about a problem and build a solution for it. The concepts of generalization and specification are identified by constructing the association between objects or classes. Since the business environment keeps changing and data is updated day by day, the association may also undergo evolution and it should be dynamic enough to cope with the turbulent market.

ii. Comprehensive Product Data Schema (PDS) is of pivotal importance for obtaining systematic data manipulation and, in turn, converting the Meta data to valuable information so that decision makers can make an informed judgment. The earlier the information is collected, the shorter time-to market and the larger market share the company can gain. A PDS can define the structure, content and semantics of XML document. This can make data retrieval easy and transform the data into reliable and on-time information which can reduce the uncertainties and disturbance.
iii. RPDS is characterized by the early involvement of different functional disciplines, some of which may not have acquired the needed information due to various reasons such as lack of communication and understanding. As such, decisions made in the early design phases of product development are often based on poorly structured, immature and outdated information (Haque et al, 2000). Very often, decisions are based on personal experience and knowledge without any systematic or structured information support. Knowledge representation is indispensable for the success of a company which wants to make further improvement of product quality in order to gain customer satisfaction.

3.4 Design Consideration

Numerous factors ought to be addressed during the design and development of the RPDS. Product design includes three critical design stage which are: conceptual design, embodiment design and detail design. Conceptual design is developed by combining physical concepts and part decomposition. Embodiment design concerns the key component design which is alternatively defined as the artifacts of the components which are: function, form and behavior. During the stage of detail design, a CAD drawing with dimensions and tolerances is prepared. It is found that the product data is continuously updated and accumulated from conceptual design to detail design. The RPDS framework must have sufficient flexibility to incorporate the iterative product development cycle.

Globalization leads to dispersed manufacturing which requires substantial collaboration and co-ordination. Take Hong Kong (HK) as an example, HK manufacturers set up the production plants in China due to low cost of labor and
premises there. The HK headquarters remains responsible for marketing, R&D and business development. The frequent data exchange between two sites is common and a universal data exchange standard is required to facilitate data extrapolation capabilities and data interchange among different platforms with business partners.

Although the revolution in product development is continuous, widespread design reuse is encouraged. For new product introduction, innovative and creative design concepts consisting of physical concepts and part decomposition has been developed. For existing products, the design concept is gradually modified by reusing the physical concept and revising the part decomposition. The physical concept embodies the way in which the product performs or provides its intended function. Product design is usually difficult to change after the product is introduced to the market as it may involve changing the customer’s habit and there is great inertia reinvested in the tooling, equipment and facilities for production. As a result, physical concept reuse occurs frequently during product development and such information should be stored in a knowledge base for future retrieval.

Product design and development is a complex process which heavily relies on specific knowledge, design principles and working experience. Knowledge dependent or independent of the domain specific heuristic needs to be taken into consideration. Artificial intelligence is comprehensively used in product design and development as it helps to formulate the knowledge representation and imitate human intelligence to solve the problem. The choice of artificial intelligence is selected to fit for the practical situation and the characteristics of the product life cycle.

A high-quality link between corporate databases and the case base is essential as recent customer requirements are stored in corporate database and design practitioners utilize the case base of the CBR system to assess the manufacturability
of the products. To have a good connection with heterogeneous databases, the case
base needs to be reconfigured and synchronized with corporate database automatically
in order to have the latest information.

3.5 Modules of RPDS

RPDS is designed to leverage knowledge such that design practitioners can
manage and respond to any change in the dynamic environment. The system
architecture of RPDS consists of three modules shown in Figure 3.1 which are: the
Object Based Product Development Module (OBPDM), the Product Information
Exchange Module (PIEM) and the Hybrid Knowledge Module (HKM). OBPDM is
used to represent the product development process and the components of the process
with an object-oriented approach. PIEM is used to standardize the mechanism of data
exchange by defining the specified PDS such that the imported PIML can be validated
by PDS. The Hybrid Knowledge Repository is used to facilitate data migration from a
corporate database to a case base where CBR is the essential part to support an
innovative and rapid product design environment.
3.5.1 Object Based Product Development Module (OBPDM)

An object-oriented design and analysis is gone through in OBPDM where all of the required detailed object classes are defined. Associations between classes are implemented and the object-oriented characteristics such as inheritance and polymorphism are applied to leverage re-use opportunities. Objectization acts as a bridge between HKM and OBPDM. All the case components are stored and reused as objects. The updated data from the corporate database is encapsulated in the case object. The inference engine is deployed such that case objects possess self-learning ability.
3.5.2 Hybrid Knowledge Module (HKM)

In the HKM, a data exchange mechanism between the database and the knowledge base is developed such that efficient data communication between heterogeneous data repositories is achieved. The case base is linked with the CBR inference engine which introduces machine learning capability of the whole system, as the CBR is able to capture past experience and knowledge for case matching in terms of product development objects defined in the OBPDM. Objectization is deployed to represent domain knowledge and to retrieve data from corporate databases following a case based paradigm. A VBA can be written for updating the database automatically so that human error can be minimized and the data integrity can be maintained. The data size of the corporate database is in terms of gigabytes. Not all data is required to be migrated into the case base where the case base is featured as organized data with valuable knowledge and professional advice for the designers. Structured Query Language (SQL) should be embedded in the VBA such that it can filter the empirical data to the case base.

3.5.3 Product Information Exchange Module (PIEM)

The PIEM is designed to assimilate immediate and accurate product information from the supply chain. The PIML is a set of XML that defines the information content from the need recognition to the prototyping stage and provides a standardized data format for data interchange internally or externally. In addition to the hierarchical static model, a dynamic model is constructed in OBPDM. Based on the static model, the PDS is defined and the incoming PIML is validated by PDS. PIEM provides a standardized way to access and manipulate the information stored in
XML documents. Using the dynamic model, java script is encoded in PIML to represent the dynamic behavior.
CHAPTER 4. DEVELOPMENT OF OBJECT BASED

PRODUCT DEVELOPMENT MODULE

(OBPDM)

Product design is a subject that has attracted many research activities. Product development starts from conceptual product design and feasibility analysis, proceeds to CAD, prototyping, tooling, production, quality control, delivery and finally product disposal. 70% of the total product cost is considered in the early phases of the product design process. Most Small and Medium Enterprises (SMEs) in HK may not consider the product life cycle from product design to disposal while they may regard the stage of product design, prototyping and tooling as product development. During the stage of detail design, engineering changes frequently occur and product designs evolve iteratively as many uncertainties are involved in transforming the abstract design concept to concrete product layout. The initial product design may predominantly be regarded as a problem rather than an opportunity. However, incremental product improvement can be achieved through iterative refinement of the conceptual design.

The main function of OBPDM is to combine different objectized components in order to perform self-learning and invoke a timely reconfigurable method. Product development goes through phases of product realization which are: identifying market needs, developing conceptual and detailed design, implementing a pilot run, and production. Cross-functional design teams facilitating nonlinear interaction are involved at each stage. Information sharing is critical for every functional department. An integrated system is necessary to facilitate the concurrent
sharing for product design information in terms of product specification, cost and schedule. Object technology is widely used for software development and causes a paradigm shift in the collaborative product design environment. Inheritance and polymorphism can be applied to the product development process. Most products are reused and redesigned by modifying the function and features in order to fulfill new customer requirements. Although the new product development needs innovative design, as a rule of thumb, the new product may have at most 40% old or original parts (Ottosson, 2002). To represent artifacts of the evolving product, objects are defined and the static relationship is identified. Each object not only possesses the attributes, but also possesses behaviors that perform the dynamic functions. The attributes and the behaviors of the product may change in accordance with the various situations which are triggered by the external environment and which alter in different periods. Therefore, temporal logic is applied to study the dynamic aspect of the object. Self-learning is also activated by generalizing the classes into superclass or abstract class and specifying the subclass with dedicated situation which is called "objectization". "Add", "delete" and "substitute" are the generic operators in the self-learning process.

4.1 Concept of Object-Oriented Design and Analysis

Object technology has become popular since early 90's and it imposes a great influence on software development. Object technology leads a paradigm shift from conventional process-based and output-based system development methodologies to the present day and object-oriented design and analysis takes advantage of system maintenance. Briefly, an object is a software program that contains related data (attributes) and procedures (behavior). Modularity helps to
reduce the dependencies of the system and thus further system evolution can be greatly enhanced. The object design method involves constructing an object-oriented model of the real world environment and mapping the model to software design. The common terms of object technology are explained below to make the concept described later more understandable.

**An object** (an instance of/a part of a class) has an inner state that is aware of the class to which it belongs. An object can interact with other objects by sending messages to each together. The receiving object accepts the message and performs the required operation. Performing an operation means that the object inspects its values, alters its own inner state or trigger other objects.

**A class** is a group of objects which share common attributes and behaviors. The class also describes the operations that the object can offer, the various activities that belong to each operation, and the different attributes that each object has. Classes are organized hierarchically and a class can inherit one or many classes. Inheriting another class means reusing the definition of that class in terms of operations, relationships and attributes.

**Abstraction** is a process of focusing on those features that are critical for the task and ignoring the detailed low-level data since there is a large amount of low-level data and it is complex to cope with. As a result, only essential information is extracted. Abstraction is a basic technique for modeling and represents a complex reality within the defined boundary for the purpose of solving a problem or understanding the operation of the organization. Three associations are commonly used in object modeling which are:

i. **part-of abstraction** (Aggregation)

ii. **a subclass of abstraction** (Generalization and specification)
Encapsulation hides both data and program code within the object. The idea is to make use of a “package” by understanding the application without knowing the theory and mechanism behind it. Encapsulation brings the advantage of stability and portability. The data and behavior encapsulated as a packet does not disrupt the nature of the packet as system change is required and it helps to maintain the stability of the system. The package can be reused and applied to various projects.

Inheritance can be explained by how children inherit features from their parents. In object-oriented design and analysis, subclasses act as children which inherit the properties, behaviors to the superclass. Once the superclass is created, the subclass can inherit the properties and behavior without redefining those from the superclass.

Polymorphism refers to the concept “to take many forms”. Polymorphism is closely related to inheritance. Normally, subclasses inherit the attributes and behaviors from the superclass whereas some behaviors of subclass may need to operate differently from the superclass. However, newly derived behavior still uses the same name as superclass. This situation is called polymorphism or delayed binding. It provides a certain extent of flexibility to handle system evolution for coping with the ever changing product development environment. As the event is triggered, the object receives a message and invokes the corresponding behavior. Each individual object can react differently to the same message. On the other hand, various objects react similarly to different messages.
4.2 Object-Oriented Modeling

Object-oriented modeling is comprehensively used in system modeling in the system design stage and Unified Modeling Language (UML) facilitates specifying the detailed operation and visualizing the information flow. Modeling is the prior stage of coding and it dedicates the (i) functional requirement, (ii) system architecture and (iii) inter-relationship among classes.

The functional requirement is extracted by studying the business process and users' working habits. The general functional requirement is decision support, data analysis and problem resolution whereas the specific functional requirements vary from each other but with the goals of increasing the efficiency and effectiveness of the work. Most CAD tools provide a drafting function by constructing a wire frame and basic surface feature and rendering function with various material textures to produce photo-realistic images. Some sophisticated CAD/CAM software such as CATIA provides more advanced features such as part stress analysis for pre-validation so the product development time can be shortened by employing a digital mock-up.

System architecture is the core part of the system and it acts as a blueprint for programmers to follow. Well-designed system architecture should be scalable, secure and adaptable. Scalability means the system allows the functions to be extended so that the information system can cater for future needs. Since most add-on components are based on existing components, new objects can be added or derived from the current object to enhance the function of the system. Security is an important issue for product data management. As a result, different groups of users have assigned authorization for accessing, modifying or deleting the data. Adaptability is that the system memorizes the implicit practice and stores it as explicit knowledge so that
knowledge based tools can give advice to the junior engineers for solving similar problems.

The inter-relationship of objects should be clearly defined since most formulae or rules may request different data input from various modules as variables or facts to find out the solution. Cross-referencing is the initial step to drive the system to determine an optimal solution with many parameters and conditions.

In this research, two techniques for object-oriented modeling are deployed. Static structural modeling allows object types to be distinguished and class, properties, behavior, association are described. Dynamic modeling describes the event and message sequence based on the temporal logic which allows the time involved and the status of object to be interpreted. Conventionally, a class diagram and an interaction diagram are applied for structural modeling and dynamic modeling in UML terminology. In this dissertation, first predicate logic and temporal logic are embedded in a static structural model and in a dynamic model respectively in order to enrich the system with reasoning capability.

4.2.1 Static Structural Modeling

An object is modeled by capturing the static structure of a system and showing the basic features of objects which are attributes and behaviors, as well as the relationship with other objects. Object-oriented modeling can be employed to model the part family which is defined as a group of related parts that have some specific sameness and similarities in terms of geometry or manufacturing process. An artifact is described as the form, feature and behavior of a product. Design-oriented part families have similar design features and geometry (form) while manufacturing-oriented families share similar manufacturing processes and operations.
(behavior). The logic of classifying parts can be employed together with constructing the object model. Part families can act as a class and the object is an instance of part families.

The product structure can be modeled by object-oriented modeling where the parent item and the child item are represented by the superclass and subclass of object model respectively. The operation of each part can be implemented by the behavior of each object or class while the form and feature of each part can be described by the attribute of the object. The detailed information of the attribute includes the attribute’s name, data type and default value. The forms of parts include the geometry, dimension and tolerance of the design parts where the data type is declared as real and the default value is the value of the standard component. Standard components are made up of common features. The common features of the parts such as slot, hole, pocket are machined by corresponding manufacturing processes which are generally classified as joining, machining, casting, bulk deformation, sheet metal working and polymer processing. A list of candidates is assessed by considering the compatibility of the material and the process, the ability of the process to produce the designed tolerance, and the availability of the process and tooling. The common features and manufacturing processes can be generalized and stored in the superclass.

In an object-oriented paradigm, operation can be query which retrieves and gets the data by parsing other objects without changing the system state; whereas modification is carried out by changing the system state. Each object has a role which corresponds to a set of behaviors of an object. Synthesizing various roles by different objects can help achieve the goal of accomplishing the task.
(I) Logical presentation of aggregation

Having determined the attribute and behavior of the objects, system analysts need to consider the association which can be in the form of recursion and hierarchy. Recursive association is an association where the source and destination are of the same object class. A typical example of recursive association is that a part consists of parts. In the view of the manufacturing process, process consists of processes. Usually, aggregation is employed for recursive association. Apart from interpreting the aggregation in form of physical units, this relationship can also be deployed in logical expression. E.g. Case consists of cases where a case can be further decomposed to solution and problem. Aggregation can be also applied such that solution/problem consists of solutions/problems. The graphical representation of this is shown in Figure 4.1.

Definition: \( p : p \in W \) where \( p \) is part, \( W \) is whole

![Diagram of aggregation](image)

Figure 4.1 Many parts are part of one product or one case consists of solution and problem

(II) Logical presentation of generation-specification

Hierarchical association indicates that the object classes are arranged from top to bottom where the object classes at the higher level are the parents of those classes lower than them. Class abstraction includes generalization and specification. Generalization and specification, which is another approach to explain the relation of
the superclass and subclasses, is used for the inheritance subclass hierarchy. Superclass generalizes its subclasses by including all the common features in the subclasses whereas subclass specializes its superclass by possessing some of the superclass instances. The common attributes shared by subclasses become the attributes of the superclass.

Definition: \(C_1 \cap C_2 = \{a : a \in C_1 \text{ and } a \in C_2\}\)

\((\forall a)(a \in C_1 \cap C_2 \Rightarrow a \in P)\)

where C is subclass

P is superclass

a is attribute/behavior

A specialized class can inherit properties and behavior from the general class.

\(C \subseteq P \text{ iff } (\forall a)(a \in P \Rightarrow a \in C)\)

where C is subclass

P is superclass

a is attribute/behavior

(III) Difference between aggregation and specification

It is necessary to understand the difference between aggregation and specification. Aggregation represents whole-part relationship and lifetime containment. For aggregation, the whole-part relation should be a concrete object and loss of any part cannot make a whole object. Likewise, specification represents a member of the super-class which is an abstract concept. Removal of one subclass does not break down the specification-generalization provided that there are more than two subclasses under the superclass.

4.9
4.2.2 Dynamic Modeling

A dynamic model is used to represent the interactive nature among the objects which emphasize the process and temporal properties of the system design. Deploying UML terminology, sequence diagram and collaboration diagrams help define interaction. UML is a visual modeling language for specifying the design requirements, visualizing the structure, sequence and interaction as well as constructing software systems.

The behavior of the entity over time during product development is dictated by the behavioral engineering. To find out the behaviors of a single object, the state chart diagram helps model the object life cycle. This describes how the object receives various messages and responds to an event as the object transits from one state to another. In the life of a product specification, the product acknowledgement form and the product brief form seem to be fundamentally or conceptually different from each other. However, those documents also capture the product description in at the early stage of product development. The information is continuously updated and the product structure evolves from the conceptual design to the detail design.

The object transits from one state to another state guided by the event and guard condition. The attributes of the objects are changed as they go from a source state to the target state. The object travels from one source state to another target state where the target state can act as the source state for the next transition. Another special situation is that one state acts as the source state and target at the same time. This is called a self-transition if the entry or exit action is fired. On the other hand, if the entry or exit action is not fired, this is called as internal transition.

As object exits one state and enters another state, the event is triggered. The event happens at a point of time and the time interval is found between two events.
From another point of view, states correspond to that time interval and possess duration. An object goes through various states. This implies that the attributes of an object have undergone certain changes due to a change in the environment or changes imposed by other objects as it moves from one state to another state. The state chart diagram shows the time sequence of events but it does not indicate the duration of states or express constraints on the evolution of the studied class of objects. Temporal logic overcomes this limitation by offering additional operators and represents this in the form of a sentence. The intuitive meaning of the operator is shown below.

- \( s \rightarrow s \) holds in the next state

- \( \Diamond s \rightarrow s \) holds either in the current state or in some future one

- \( \Box s \rightarrow s \) holds in the current state and in all future ones

where \( s \) is a statement

**State:** A mode of condition of being

**Transition:** (1) The act or experience of moving from one state to another (2) the condition is fulfilled which causes the object to change and puts it into another state.

The notion of "change" occurring in the object is formalized by the following definition.

\(-\text{saved}(CAD) \land \lnot (\text{saved}(CAD))\)

**Event:** (i) Events are issues occurring in a certain space and time (ii) An Event as a transition (iii) Events as occurrences over intervals of time.

The structure of time related to product development can be explored generally by finding the answer to three questions:-

1. Is time discrete or dense? If it is defined that the facts vary on a time scale consisting only of whole numbers then one can use a discrete time model while if it
varies on a timescale consisting of real numbers, then one can use a dense time model (Pani and Bhattacharjee, 2001). However, the definition of the timescale may use a deterministic role for the time model. E.g. Half year/6 months can also be used as a timescale. It can be regarded as one that varies on a month scale consisting of whole numbers, and then one uses a discrete time model. On the other hand, if a year time scale is used, it takes half a year (real numbers) for any element more than two or above with the timescale as a dense time model is employed.

2. Is time bounded or unbounded? A similar question is raised about whether time is finite or infinite in extent. If it is believed that there is one starting point and one end point of the time line, time is stated as bounded. Otherwise, time is unbounded as every time is succeeded by a later time or preceded by an earlier time. For product development, the product needs to be delivered at a specific time in the future. In system development, time is bounded for product development activities.

3. Is time linear, branching, parallel or circular? The gross topological properties of time give us an idea about the implementation mode of process. Linear seems to be carried out sequentially while parallel are process undertaken concurrently. Product life cycle is treated as circular and product development cycle including product introduction, growth maturity and sales decline is one in which the processes/status are repeated periodically. In a macro view, the process occurs cyclically. In the micro view, the process is customized to adapt to the changing situation even if it follows the same sequence of stages.

In this research, a finite set of basic predefined types including usual types as well as time type, is employed in the proposed TRML. The domain of time is the set of \( TIME = \{0, 1, \text{now}\} \), isomorphic to the set of natural numbers. 0 is the starting point while now is the current status. Times go by and time increments one by one. In

4.12
realistically, the properties of object and behavior will not freeze at a particular state. Each object in the product development cycle interacts with each other. The behavior of one object may affect the behavior or other object and induce changes in the attributes of the objects. E.g. the mobile phone user would like to change the case of the mobile phone frequently. This is regarded as customer behavior. That leads to mobile phone manufacturers producing mobile phones with changeable cases which are a new function or property of the new mobile phone. Behavior goes along with event and event is usually found to be valid within a particular time interval. As a result, temporal logic or the concept of time should be included in object modeling for the dynamic environment. Sequence diagrams depicting the dynamic behavior of elements and the introduction of time units under the class role help identify the interactions of the element over time. Figure 4.2 shows some typical activities involved in the Engineering Change Note (ECN) lifecycle where the procedures start with the creation of ECN to store items of change (Dale, 1982). The objects involved include the engineering department, inventory manager, product engineer, industrial engineers and engineering change notes. As time is defined as starting from 0 and each interaction is assumed to be one unit, the time duration can be found under the class. E.g. the engineering department has 3 interactions with other elements and each take 3 units of time. As the engineering department is the initiator, the time interval starts from zero and the finishing unit of time is 3 at the end of the third interaction. The coordinator possesses the first sequence of activity (forward package) which is represented by the second arrow from the initiator with the engineering department such that the initial unit of time is 1 and the finishing unit of time is based on the last interaction (sequence of activities) of the object. Since there is only action for purchasing, the time interval is 6-6 so that the action completes instantly. The
sequence diagram with time duration shown in Figure 4.2 clearly depicts the time that
the object starts and the time it accomplishes the tasks.

![Sequence diagram of engineering change management](image)

Figure 4.2 Sequence diagram of engineering change management

4.3 Timely Reconfigurable Method Invocation (TRMI)

TRMI is derived from the advanced concept of object technology which is
polymorphism where polymorphism is the extension of inheritance. This is useful for
reusing the code through inheriting the property and function from the superclass
while allowing a certain flexibility to handle individual needs.
4.3.1 Declaration of TRMI

The TRMI is performed to achieve the goal of responsiveness to the external stimulus by reconfiguring the various methods to accomplish the same operation. As one object initiates a message, the format is stipulated in a message signature which specifies the name of the method to be executed and the parameters to be included. The same operation may apply to different classes and such operation is called polymorphism which refers to the same operation taking on different forms in different class. Conventionally, a method is the implementation of an operation for a class. However, the authors find that an operation needs to be implemented by synthesizing several methods with a specific pattern under different conditions which is called as "TRMI".

Definition 1: For $M \in D$ the set of all configurations of $M$ is denoted by $\text{Conf}_M$ where $\text{Conf}_M$ consists of a finite set of methods with partially conjoint sets of operations $a$.

Proposition 1. Given $M_1, M_2 \in D$, $M_1 \circ M_2$ is correct if and only if all of the following conditions hold

Condition 1. $M_1 \cap M_2 \neq 0$ and $M_1 \cup M_2 \neq 0$

$\implies$ $M_1$ and $M_2$ is not mutually exclusive and $M_1$ is not equal to $M_2$

Condition 2. Name of $M_1$=Name of $M_2$ where the final operation is the same. i.e. $M_{n1} = M_{n2}$.

$$M_{n1} = \bigcup_{i=1}^{k} a_i, M_{n2} = \bigcup_{i=1}^{k} a_i$$
4.3.2 The logic of TRMI

Logic is designed to specify reason and represent the algorithms, programs and systems. The reasoning mechanism is the essential part to support knowledge discovery. Reasoning is instrumental in the product design process where reuse of previous experience and creation of new ideas are the two key factors to support an innovative and rapid product design environment. In Figure 4.3 the mechanism/flow of change in product specification is in form of function, geometry and material. To simplify and give a clear picture of design, those three kinds of design change are regarded as mutually exclusive. Geometry modification includes changes in dimension, tolerance and surface finish. Having considered the technology possessed by the manufacturing firm, engineers determine the optimal solution by considering the cost, customer requirements as well as the defined geometry parameters that can be fabricated by the existing equipment and tooling. Geometric changes can be performed by modifying the STEP data or editing the graphics file directly by using the CAD/CAM software. Modification of surface finishes or production of goods with new materials needs the creation of an engineering change request. Then the detailed engineering change data should be sent to the responsible person to perform an evaluation of the design change. As new materials continually emerge, the new product may adopt those new materials to achieve some specific functions. E.g. some goggles are made by titanium alloy that is sturdy and allow hingeless and screwless connection of lenses.
Figure 4.3  Process flow of TRMI for product development
4.3.3 Static Representation of TRMI

Design change is common during product development and the cause of design change is usually triggered by two parties. The first party is the external customer who changes the requirement in accordance with the market trend or other competitors' emerging product feature. The second party is the internal customer, that is, the technical department, who may suggest design modification based on the technical constraints such as constraints imposed by production capacity, tools and equipment. The requested design tolerance and surface finishes are determined by considering the compatibility of the material and process, the functionality of the machine and tooling. The type of design change can be generally classified into the modification in material, geometry or function. In the class diagram of Figure 4.4 the product specification contains the detailed information of the part such as part name, part number, description and those common properties are passed on to the subclass. As customer and engineering department usually request functional changes at the beginning of the product development cycle, the change of product specification will be directly related to the functional modification. Product specification has an instance in the functional class which uses the program code from the product specifications by inheriting the properties and behaviors from the product specification. When design change is triggered due to the modification of material or geometry, the class of geometry and material use their relative program code to implement the design change rather than inheriting the properties and behaviors from the product specification as functional change. In the class diagram, it is found that the duplication of change under the geometry and material class it is regarded as polymorphism. The duplication term “change” found in superclass and subclass implies that the change of material of function is quite different from the change inherited in the product specification.
However, it is realized that the behavior "change" involves a sequence of steps, to perform such "change" which cannot be illustrated in the class diagram.

Figure 4.4 Object model to illustrate the concept of polymorphism
4.3.4 Dynamic Modeling of TRMI

As mentioned before, there are generally three kinds of changes which are related to geometry, material and function. Figure 4.5 illustrates geometrical change. Event B1 is initiated by the designer where it causes the transition from State1 (Saved) to State 2 (Retrieved). Event B2 also comes from designers and event B3 from the CAD system which dictates that several sources for the event type. Events correspond to the messages (that is, function or method calls) that form a communication channel between objects.

Entry action and exit action for each state (to be performed as the CAD file enters and leaves each state) have been written inside the boxes for the state. The actions, which are atomic behaviors that go with state or a transition, include version and date update. The draft object may include CAD file name, designer ID or CAD number so that the system knows which designer prepared the draft.

When the designers request a CAD file, the CAD file will go to “retrieve” state. As it performs the specified exit action, it updates the version of the CAD file. As the CAD file enters the analysis state, the entry action “archive the draft” is carried out. In analysis states, there are two possible responses to the event B3 (PDM system saved the CAD file). The design may need to analyze the design with simulation tools and those designs with potential defects require further modification. Designs without potential defect will be saved in the PDM system. The two conditions mentioned above are mutually exclusive. This is regarded as a guard condition for the analysis state as that may prohibit the transition if the condition of the product design, without potential defects, is evaluated as FALSE.
Event B5 is a self transition which fires the exit and then enters the transition of the state (which is acting as both source and target) as it leaves and re-enters that state (Brown 2002). As the old product model is obsolete, the new product model may use the same product name or product ID but with the updated version. The version and date of creation is different because the new product model is saved in the system. The attribute of is version is updated and the execution occurs for a replacement and the entry action is also fired.

![Diagram showing state of transition of CAD modification]

Figure 4.5 State of transition of CAD modification

4.3.5 Case Example in the Product Development Process

In practice, the design change is complex and complicated as the three kinds of changes are interrelated and one design change may trigger change of others where
a chain of reaction may occur. To implement the design changes, the various departments involved implement the same operation differently. A list of methods is shown below.

A. Evaluate production capacity
B. Check production schedule
C. Make decision (accept or reject design change)
D. Notify the department of production and material control
E. Negotiate with marketing department
F. Implement the engineering change processes
G. Delete design change

Situation 1: There is not enough production capacity for the design change. The new or revised design feature needs to be deleted and this behavior is shown in the superclass (Design Review), and “delete design change” is inherited to the subclass, (Infeasible Design Change). Therefore, methods are carried out under the following sequence which is A-B-C (reject)-E-G.

Situation 2: This is an instance of the feasible design and it is located in the “Feasible Design Change” and runs the “delete design change” operation after the design change process has been completed. Therefore, methods are carried out in the following sequence which is A-B-C (accept)-D-F-G.

The same declaration of the behavior, “delete design change”, with different meaning is not allowed to occur in the past procedural program as it cannot distinguish the semantics of “Delete Design Review Detail”. However, the object-oriented program allows “Polymorphism”. The subclass version of operation “Delete design change” shown in Figure 4.6 overrides the version from the superclass as the system
uses the version of operation from the subclass in preference to the superclass version, for that subclass and all its descendants (Brown, 2002).

Figure 4.6 Object model of dynamic objects

This chapter has described how object modeling is applied in Object Based Product Development Module (OBPDM) for system modeling. The experience with TRMI has advantageous implications as TRMI is the extended concept of polymorphism where polymorphism is a critical and promising feature of object technology. The following chapter will exploit a mechanism to transform data to a knowledge base to allow users to have instant access to the CBR system for recasting reasoning.

4.23
CHAPTER 5. DEVELOPMENT OF HYBRID KNOWLEDGE MODULE (HKM)

The first part of this chapter introduces a generic data interchange mechanism and the second part of this chapter illustrates how to make use of CBR to achieve efficient knowledge acquisition and update information retrieval. Corporate knowledge is contextualized and stored as experience in the form of cases in a case base which needs to be updated with the latest information from the database. The broken linkage between corporate databases and the knowledge repository inevitably undermines control over integrity. The main component of the HKM is characterized by its capability to handle two main types of data from the two dissimilar database systems, the corporate database and the knowledge repository (case base). CBR, which is characterized by utilizing the specific knowledge of previous experience and concrete problem scenarios (cases), is widely used in problem solving (Watson, 1997). A new problem is resolved by finding a similar case and adapting it to a new situation. The adaptation is processed by sustained and incremental learning. The system of CBR for problem solving is comprehensive only in certain routine designs that have a clearly defined design process. Current efforts at maintenance of CBR systems are generally devoted to the control of case base growth and to the improvement of accuracy of case knowledge (Cheetham, 2000).
5.1 Mapping between the Corporate Database and the Case Base

HKM constitutes a formal and logical specification of the components, which are the interface for the relational database, the migration program and the interface for the case base. The migration program is an intermediate component which contains VBA and SQL where SQL is used in querying, updating, and managing relational databases while VBA is used to update the data dynamically. The knowledge workers may input the new data or update data through commercial application software and this data is stored in the corporate database. HKM plays a crucial role in detecting the new and updated data in the relational database. As there is new data or updated data, HKM will then export the data in text format which can in turn be the source of the CBR system. If there is not any updated data, the stored data in the case base is retrieved and displayed for the end-users. The design practitioners can acquire the knowledge they need from the case base and it helps them to solve the problems during the early phase of product development. The data in the cooperate database may be modified frequently due to the ever-changing environment.

5.1.1 Corporate Database

The corporate database is normally the mainstream system used in companies to form data tables such as a sales record data table. As the corporate database is characterized by its efficient searching mechanism for data stored in the fields of a data
table, the data in these fields can be managed efficiently. Using the relational data model mechanism, users can locate the required data quickly for further processing. Most of the relational database tools such as Microsoft® Access provide this search facility. In brief, relational databases are the most common data models with the advantages of (i) being well-developed with adequate supporting tools and literature for those who intend to build customized relational database applications, (ii) having relatively less RAM memory to operate than other data models such as object data model, and (iii) being reliable in handling alphanumeric data files (House, 1991). As shown in Figure 5.1, the relational database system contains a number of data tables, which are inter-related through the 1 to many or many to 1 relationship. However, this relational data structure is not designed to handle complex data such as images or animation; and it cannot handle data objects, which includes attributes (data fields such as age, address) and behavior (program code to define certain functions or procedures).

CASE NAME: Upper case small
Description: Decrease tends to have a small size product
Solution: The smallest product in the past
Product ID: 99998
Size: 123 x 56 x 79

FST has started to conduct a research named "Be Small" in 1/1/2001 and they now use laser drilling to achieve surface mounting technology to develop smaller size circuit board. It is expected to complete in 12/3/2002. The smaller the case will be created.

Figure 5.1 Correlation between case base and relational database
5.1.2 Knowledge Repository

CBR tools normally incorporate their own knowledge repository to suit their functions. However, the majority of the corporate databases support relational data communication. According to a recent survey (Haque et al, 2000), there are four main CBR tools: The Easy Reasoner, CBR-Kate, CBR-Works and Recall. All of them, to a certain extent, support the data transfer feature with relational database systems. This is an essential feature for most of the knowledge database tools as the software solution will not be useful if compatibility with the corporate database system cannot be achieved.

As shown in Figure 5.1, the CBR knowledge storage system comprises individual files with a free format, which are basically in the form of text files. Theoretically, any character, word, or combination of these, can be found, changed, deleted, inserted or replaced using the text editor. The files may be under different categories such as product design, plant layout, etc. In normal cases, the knowledge base, which is an integral part of the CBR tool, is designed to fit specifically the unique system configuration of the software. Although most of the CBR tools claim that they are equipped with the data transfer capability, direct data transfer cannot be achieved due to the dissimilar data formats of the relational data format and the free format for the CBR.

Obviously, the information exchange capability of the entire operation can be enhanced if the case base, which contains case problems and solutions, is seamlessly integrated with the corporate database, which contains data tables with records. As shown in Figure 5.1, HKM is of a ‘dynamic’ nature, which grows with experience over time. As a result, this system allows users to build up knowledge by inputting new data to the module.
5.1.3 The Dynamic Data Exchange Mechanism between Knowledge Repository and Relational Database

The dynamic data exchange mechanism constitutes a formal and logical specification of the components, which are the interface for relational database, the migration program and the interface for the knowledge repository. The migration program is an intermediate component which contains VB and SQL where SQL is used for querying, updating, and managing relational databases while VBA is used to update the data dynamically. For example, the SQL statement "SELECT (select list) FROM (anywhere a table name, query name, or field name) WHERE (condition argument expression)" is used to formulate a special relation between the case base and the corporate database.

Figure 5.2 shows the data flow of updating the case base from the relational database. The new data or updated data is input through commercial application software and the data is stored in the corporate database. The dynamic data exchange mechanism plays a crucial role in detecting the new and updated data in the relational database. As there is new data or updated data, the dynamic data exchange mechanism is processed by exporting the data as text format which can in turn be the source of the CBR system. If there is not any updated data, the stored data in case base is retrieved and displayed for the end-users. The design practitioners can acquire the knowledge from the CBR system and it helps them solve the problems during the early phase of product development.
Figure 5.2 The flow of updating the case base with corporate data

The data in the corporate database may be modified frequently due to the ever-changing environment. The dynamic data exchange mechanism is designed to detect any changes in the corporate database and then update the knowledge repository. A special kind of stored procedure, a "trigger", can be created to cascade changes through related tables in the corporate database. Suppose there are some changes in the "Project" tables. An e-mail is sent to the corresponding interested people, JohnS.
Use of "trigger" for sending a reminder e-mail message:

```
CREATE TRIGGER reminder
ON Project
FOR INSERT, UPDATE, DELETE
AS
EXEC master..xp_sendmail 'JohnS',
'The date of the project has been changed.'
GO
```

When the alert message is sent out, it updates the text file with the text index and replaces the project date with the updated field from the relational database. It is found that the solution to a case may contain various data from different tables and only the good mapping of the data can provide accurate and meaningful information for the users. The following example shows that dynamic data exchange mechanism makes use of VB script "SearchString" and replaces the old data with the new one.

**Situation:** One project engineer wants to know how to schedule the project such that the idle time can be minimized. The CBR provides him with a solution, which displays the arrangement of various projects, starting dates and completion dates. In particular, the data in the relational database should point to a specific position in the solution of a text file such that it can convert correctly to the relative field in the case base. It can be imagined that the confusion occurs when the starting date field of the relational database is exported to the wrong field, for example, the field of "complete date" in the case base. Mapping the data with an index can reduce the occurrence of such errors. The following VB script illustrates how to convert the data correctly from corporate database to knowledge repository.
GUM may start to search the target ProjectID within the text 1 which is the text file of case base. MyPos = Instr(3, SearchString, SearchChar, 1) means to have a textual comparison starting at position 3.

As the dynamic data exchange mechanism successfully maps the field in relational database to the knowledge intensive case base, data transformation starts. The critical process to export the updated data from the relational database to CBR system is to change the data into a neutral format known as a text file. This is because most CBR systems are able to read a text file and convert it into the case format. The text file in case format contains the delimiters such as “Case Name:”, “Description:”, “Question:” and “Solution:”. CBR software searches the delimiters which indicate the position of data in text files such that the dynamic data exchange mechanism initiates to update the data with a new set of fields in the corporate database.

Having inserted those delimiters into the text file, CBR can read the relevant information and write the data to the case base. Different CBR tools may have their own operations to convert the text content to cases. Case Advisor 4.12 for example, simply requires users to click the button “converting text to cases” and the CBR system can “dig out” questions/query from the case information. While extracting the cases from the text file, Case Authoring Module of Case Advisor 4.12 also extracts key words and key

5.8
phrases from the cases. If the case can be extracted, the associated questions will be created from key phrases and words.

5.2 Objectization of Case Based Reasoning

RPDS performs “self-learning” by accessing updated data in the corporate database while the case base stores the past cases of product development in which multifunctioning departments are involved. Cases are represented as collections of objects and the slots of cases are described as the attributes or behaviors of objects. Object classes are structured in a class hierarchy and the subclasses (child cases) inherit attributes (slot) from the superclass (parent case). When adaptation has been successfully implemented, RPDS only stores the superclass rather than the subclass. All aspects of the user interface are dynamically created on demand, specialized for the situation, the specific users, and the interface devices that they choose to use. Since the superclass is the generic node that can generate the subclass for specific uses. New cases which have only differences in value need not be stored in the knowledge repository whereas those new case that have new attributes or behavior from the original case will be retained. The objectization goes through the steps shown in Figure 5.3.
5.2.1 Case Representation

Knowledge representation can be generally classified into four logical representation schemes: logical representation schemes, procedural representation schemes, network representation schemes and structured representation schemes (Mylopoulos and Brodie, 1989). Inheritance, which is an important feature of almost all network and structured representation language is implemented in this research. The cases are generalised from a superclass and categorized by the K Nearest-Neighbour (K-NN)
algorithm. The related cases are grouped together while a unique case name or index is
given to each case for identification.

5.2.1.1 Generalizing the Case (Objects) to Form an Abstract Case (Abstract Class)
or a Generalized Case (Superclass)

Generalization is the first step for "objectization". To generalize several
subclasses (specified cases) to superclass (generalized case), a similarity assessment
among sub-classes is conducted. Having defined the common features of sub-classes,
K-NN is used for categorization. The similarity between the inter-class and intra-class is
measured. As a result, identifying which class or superclass an instance or a subclass
belongs to, is important. Similar subclasses can be grouped horizontally or vertically and
a representative name can be given to the superclass.

5.2.1.2 Case Categorization Based on the K-Nearest Neighbour Approach

Various learning based approaches have been applied to categorization.
Bergmann and Stahl(1998) propose similarity measures for object oriented case
representation that make use of class hierarchy, relational attributes and flexible local
similarity measures for simple attributes. Kwon and Lee (2002) derive a scheme for web
site classification based on the K-NN approach. K-NN is suggested by authors as one of
the categorization techniques for classifying the cases in the case repository. Three
processes are needed in order to proceed:

(i) Identify whether it is possible to attach instance X to class C,
(ii) Realize which instance of X can be attached to which specific sub-classes of C

(iii) Recognize the amphibian (a case that is difficult to categorize as it has attributes-slot from more than one superclass). Such cases are temporarily disregarded.

During classification, it is assumed that there is already a class Cj which has Oi as one of the instances in class Cj. The likelihood of Object Ox in Cj is estimated by summing the weight of the category of the K-NN object as follows:

\[ P(Cj|Ox) = \sum SIM(Ox, Oi) P(Cj|Oi) \quad \ldots \quad (1) \]

where Ox represent the test object

Oi represents an object which is already present in class Cj,

SIM(Ox, Oi) represents the similarity between the test Object Ox and training Object Oi,

\[ P(Cj|Oi) \in \{0, 1\} \] represents the classification for the Object Oi with respect to category Cj or sub-class Cj

To determine the similarity of object Ox and Oi, assessing the properties of both objects is the most convenient method.

\[ SIM(Ox,Oi)=\Phi(sim_{A1}(x,A_i,\ldots,i,A_n),\ldots,(sim_{A1}(x,A_i,\ldots,i,A_n))\ldots\ldots(2) \]

where x.A1 represent the first property of the test object

i.A1 represent the first property of the object which is already present in class Cj.

x.An represent the nth property of the test object

i.An represent the nth property of the object which is already present in class Cj.

5.12
By substituting equation 2 into equation 1, the following equation is obtained:

\[ P(C_j|O_x) = \sum (\Phi(sim_{A_1}(x,A_1,i,A_1), \ldots, sim_{A_n}(x,A_n,i,A_n)))P(C_j|O_i) \ldots \ldots \ldots (3) \]

For those results which are 1, i.e. \( P(C_j|O_x) = 1 \), the object \( O_x \) is under the class \( C_j \).

For those results which are 0, i.e. \( P(C_j|O_x) = 0 \), the object \( O_x \) is not under the class \( C_j \).

For those results which are 0.5, i.e.\( P(C_j|O_x) = 0.5 \), the object \( O_x \) is amphibian which means it cannot be determined whether it is under \( C_j \) or not. The formula should be applied to other classes and check whether the result of \( P(C_n|O_x) \) gives a higher value or not.

5.2.1.3 Grouping

Once a group of cases under the same category is identified, grouping those similar cases is required. For those similar cases, the boundary of grouping similar cases must be established. Such an arrangement is shown in Figure 5.4. Consider whether group H and I form a D' or group H, I, D, E together to form B' . If the grouping boundary is wide, it helps simplify the case hierarchy. However, simplification can lead to difficulties in finding specific information. Therefore, equilibrium needs to be maintained in order to ensure the system is specific enough to access the required information but simple enough to facilitate navigation through the case base. The categorized cases should be mutually exclusive but not collectively exhaustive.
5.2.1.4 Representative Name for the Generalized Case

Once similar cases are grouped into category or similar cases are grouped to form a generalized case, a unique name should be assigned. "Name" can be regarded as the attribute of the case. Various cases can contain the same slot (attribute) with the same value. System analysts need to exercise caution in selecting slots (attributes) for case indexing since the index should be unique and one index should not be assigned to more than one case, otherwise it may lead to confusion. The criteria for the index should be abstract enough to allow for widening it for the future use of the generalized case and concrete enough to be recognized in the future.
5.2.2 Case Retrieval

HKM first determines the similarities between stored cases and new problems. Inductive retrieval is used to determine the similarity of the case features between the stored case and the new problem. For retrieving an appropriate case, it is important to pay attention to the semantics of the case as it helps to enhance the efficiency of case adaptation.

5.2.2.1 Case Selection

The retrieved case should have an artifact which has been successfully adapted to solve a problem similar to the one that now needs to be solved. If a similar case is retrieved, the adaptation process can be applied again and the next procedures can be carried out. The similar case object can be grouped to form a generalized class which can be an abstract case. The cases that contain similar artifacts will be grouped together. Artifacts consist of functions, forms and behaviors which are three critical representations of customer requirements during the product design phase. Identifying the common artifacts among instances helps in understanding the case more thoroughly.

5.2.2.2 Inductive Retrieval

Induction is an effective data mining technique that extracts decision knowledge directly from a large database or knowledge domain and represents it in the form of a decision. In objectized CBR, the case base is analyzed and categorized by the K-NN algorithm to produce a decision tree that classifies the cases. In product design, for
example, good product design should minimize the total number of the parts, maximize
standard components and avoid secondary operations. Table 5.1 shows four product
design cases collected from the past:

Table 5.1 Product design cases extracted from the case base

<table>
<thead>
<tr>
<th></th>
<th>Product design</th>
<th>Number of standard components</th>
<th>Secondary operations</th>
<th>Total number of parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>Good</td>
<td>4</td>
<td>N</td>
<td>20</td>
</tr>
<tr>
<td>Case 2</td>
<td>Very Bad</td>
<td>8</td>
<td>N</td>
<td>60</td>
</tr>
<tr>
<td>Case 3</td>
<td>Very good</td>
<td>6</td>
<td>Y</td>
<td>30</td>
</tr>
<tr>
<td>Case 4</td>
<td>Bad</td>
<td>3</td>
<td>N</td>
<td>50</td>
</tr>
</tbody>
</table>

A decision tree shown in Figure 5.5 is formed by analyzing the case stored in the case
base. The predictors are set as follows:

(i) Total numbers of parts <40  (ii) Number of standard components>5  (iii)Secondary
operations.

Figure 5.5 The generalized decision tree

5.16
Assume there is one query [(a) Total number of parts = 25; (b) Number of standard components = 6; (c) Secondary operations = Yes], it is regarded as a good product design based on the above generalized decision tree as the query case is similar to case 3.

5.2.2.3 Comparison by Paying Attention to the Semantics of the Case

CBR system retrieves similar cases by comparing the slots of the case which are the attributes or a description of the case. Identification of discrepancies is the first step in recognizing the similarity between two cases. They can be distinguished by the number of slots, slot values, slot types and structure. Inductive retrieval and nearest-neighbour algorithm, which are common retrieval techniques used for CBR, implements comparison among sibling groups. Sibling group are the cases of the same level in the hierarchy. The retrieval process includes identifying features, searching, initially matching and selection. The retriever needs to identify the problem description and to distinguish the noise description from the useful one. The CBR system can disregard unknown features which are treated as non-existent or it can set a null value for that feature. Case matching is performed in order to identify similarities and discrepancies among cases.

Similarity assessment is a knowledge intensive processes which is needed in order to identify the semantics of the problem. A similarity matrix usually measures the surface similarity of the problem or case. If CBR is used only for queries and the retrieved cases need not be adapted, the retrieved case is expected to be very similar to the query case. However, as the retrieved case needs to undergo further adaptation, a higher surface similarity in features does not imply that the case is easy to adapt. This is because it is
more straightforward to impose adaptation rules onto a generalized case rather than onto a specific case.

According to the Inventor's Paradox, Polya (1949) observed that it is often easier to solve a more general problem than the one at hand and then to use the solution of the general problem to solve the specific problem. E.g. Product design should be aesthetically pleasing and this can usually be achieved by using symmetrical visual effects. How to make the product (PDA V888) have aesthetic value is a specific problem. To generalize the problem, symmetry usually gives customer a sense of aesthetic pleasure and the generalized solution for the problem is to mirror the object by finding the middle line. The generalized solution can also apply to specific solutions such as mirroring the one part into two by finding the centre of symmetry. It has been mentioned before that most CBR systems can only match surface similarity and neglect the semantics of the case. It is necessary to know the semantics or rules to adapt the case in order to fit the ultimate purpose or fulfil the predefined requirement. It is not easy to find out the deep semantics as this involves special terminology. The CBR system allows users to define words in a thesaurus in order to increase the efficiency of the retrieval process. The CBR system should not retrieve similar cases by comparing the string input with the descriptive data of the stored cases. Suppose there are two cases in the case base where one is about a non-symmetrical pentagon and the other one is a symmetric heptagon. The query is about finding out the way to divide the pentagon equally. If the retriever only considers the surface features, the retriever may find that the non-symmetrical pentagon is more similar to the query case since both are five sided polygons. However, if adaptation needs to be carried out after retrieval, the second case (symmetric heptagon), which is more relevant
than the first case, is easier to adapt to solve the problem of dividing the pentagon equally into two equal parts. It is because the retriever knows the semantics and criteria of dividing the polygon into two equal parts symmetrically.

5.2.3 Case Adaptation

Case adaptation is applied if the new problem solving operation is found to be different from cases in the domain. Case reasoner of HKM retrieves the most similar case and transforms the existing solution in an operation which is suitable for the current problem. Adaptation looks for major difference between the retrieved case and the current case and applies rules that take the difference into account when suggesting a solution.

5.2.3.1 Backward Chaining from the Target Object to Superclass

Adaptation is regarded as the self-learning process of case based reasoning. Specification is required in order to customize the generalized solution under specified case situation.

Case 1: square can be divided into two halves by drawing a middle line or diagonal.

Case 2: a heptagon can be divided into half by drawing a line from the vertex to the middle of the opposite line.

A rule is derived by observing the above cases.

Rule: If the polygon is symmetric,

Then it can be divided into half by drawing a line at the symmetrical point

The main feature and semantic feature that fired the rule is “symmetry”. The symmetrical pentagon is an instance of a symmetrical polygon which means that the rule can be
applied successfully in this situation. Subclass (symmetrical pentagon) inherits not only the behaviors and attributes but also the rules from the superclass (symmetrical polygon). Specification, which customizes the general solution to fit individual needs, can be achieved by parameter adjustment, reinstanitation, derivational replay or model guided repair. The above example has employed model guided repair as the problem domain which is analyzed and understood. The old solution is adapted to solve the new problem by applying the adaptation rules. It is found to be feasible to apply the rule by realizing the relationship of the specific case and the generalized case. In the generalization and specification process, Figure 5.6 shows that the cases arranged in hierarchy are first generalized to form a superclass and then customized to a case fit for a specific situation.

Figure 5.6  Generalizing the case to form a superclass and deriving a new solution by specification
The specific data of cases is kept in the working memory and the inference engine searches a state space in two directions that is either goal driven or data driven. Most likely, the query case provides crucial data and information for retrieval and the forward chaining rule is employed to find the solution during the adaptation process. However, if the query case contains loose information but needs to get the predefined sub-goal for achieving the terminal goal, backward chaining is deployed to verify the hypothesis. As the case can be solved by using the adaptation rules that draw a symmetric line to divide the polygon, it can be concluded that the polygon is symmetrical. HKM may select a different searching method based on the information given by users.

5.2.3.2 Deontic Logic in Case Adaptation

Case adaptation is carried out when the generalized case is specified and adapted to the new problem. In this research, deontic logic is applied for case adaptation to resolve constraint satisfaction problems during product design. Deontic logic, which concerns reasoning about the norms including Permission(P), Forbidden(F) and Obligation(O), has been picked up by computer scientists and AI researchers. The semantics of the symbol is depicted as below.

\( \neg \) for negation

\( \Rightarrow \) for material conditional

\( \equiv \) for material biconditions

The von Wright-type system (1968) is employed to clearly define permission, obligation and forbidden in the following axioms and rules.
KD0: All (or enough) tautologies of Propositional Calculus

KD1: O(a⇒b) ⊃(Oa⇒Ob)

KD2: Oa⇒Pa

KD3: Pa≡¬O¬a

KD4: Fa≡¬Pa

KD5: Modus Ponens: a, a⇒b / b

KD1 is known as K-axiom and KD2 is called the D-axiom. KD2 states that obligation of a implies permitted of a under material conditions. KD3 states that permission is the dual of obligation and KD4 states that forbidden is not permitted.

Good product design should include not only aesthetic and functional aspects but also those of manufacturability and the economics of production. Some intelligent design tools should embed the following design rules and guidelines to assist design engineers to find the optimal solutions with regard to quality, cost and time. The deontic axioms and theorem provide a formal system to represent the design rules of the design process. This approach can be applied to the management of resource constraints within the product development process. Using the pre-defined design guidelines, it is possible to study the criteria of design activities and provide a context for human actors. Decision-support can be provided to modify infeasible designs created by junior designers that further reduce the possibility of production or create assembly problems.

This formalism is demonstrated as fulfilling a design requirement from a realistic situation and serves to demonstrate the practicality of this solution approach. The design principle in the product lifecycle includes:
i. **Standardization**—standard materials and components should be used in order to avoid undesirable interactions due to variation from model to model in a product family. Using standard components can minimize design time and cost.

ii. **Specification**—liberal tolerance and dimensions should be specified for ease of analyzing the proposed design with available manufacturing processes. Clearly-stated tolerances and dimensions can avoid misunderstanding occurring in the production process and thus reduce the non-conformance aspects of the product and enhance the product quality.

iii. **Synthesis**—Raw material is converted to finished/semi-finished products using a value-adding manufacturing process. An integrated product process design and avoidance of secondary operations can shorten the product development time. Making use of standard components can save equipment setup time and shorten the production time. By applying the deontic formalism to the design principles, a set of deontic rules is generated and presented as follows:

Principle 1: Standardization

\[ KD1: O\{(use, standard material) \supset (use, standard component)\} \supset \{O(use, standard material) \supset O(use, standard component)\} \]

Principle 2: Specification

\[ KD2: O(state, liberal tolerance) \supset P(state, liberal tolerance) \]

Principle 3: Synthesis

\[ KD3: P(use, processible material) \equiv \neg O(use impossible material) \]

\[ KD4: F(perform, secondary operations) \equiv \neg P(perform, secondary operation) \]
Another example of the application of deontic logic related to rule-based adaptation is represented in Prolog which uses the described predicates. Standardization, tolerance and dimension specification as well as the manufacturability are stipulated as obligations for design practitioners:

"Produce the goods on time with the quality acceptable by the customer within the conformance level, on the delivery date or within the period, at a reasonable price."

This clause is represented in CBR by the following rules:

oblige_act (produce, manufacturer, DesignPrinciple, on time, delivery date, quality, conformance, reasonable price) IF
extra (production, manufacturer, DesignPrinciple, Manner) AND
fact (principle, DesignPrinciple, Principle) AND
in_list (Principle, ["Standardization", "Specification", "Synthesis"]) AND
fact (quality, DesignPrinciple, Quality) AND
fact (conformancelevel, DesignPrinciple, Conformance) AND
fact (date_of_delivery, DesignPrinciple, Date) AND
no (exception (production, manufacturer, DesignPrinciple)).

5.2.4 Case Maintenance

Case retention has attracted the attention of researchers because cases accumulate exponentially when newly adapted cases are stored for further retrieval. New cases are derived from the existing cases, evaluated and subsumed under the
superclass. The superclass rather than the new subclass is stored in the case base. The system provides guided questions to let users decide whether the newly derived sub-class cases or the original sub-class cases should be retained in the case base. The super-class will not be deleted except if the administrator of the CBR system decides to revise the structure of the case base. Guided questions lead the users to determine whether the case is one of null adaptation, parameter adjustment, reinstatiation, derivational replay or model-guided repair. If the retrieved case has no adaptation applied to it, the retrieved case does not need to be saved as a new case. Otherwise, it may become a duplicate record. Retrieved adapted cases with the same parameters but new parameter value, do not need to be saved since the superclass/generalized case is stored and the feature/property is inherited to the subclass/specified case. The new parameter value can be adjusted with the adaptation rules. Users can retrieve the case by accessing the generalized case and attuning the parameter value. Reinstantiation is used to derive new features from the old solution if the adapted case contains some new feature which is not possessed by the generalized case/superclass. Those newly adapted cases need to be stored and periodic reviews are required to induce the common new feature from the specified case to generalized cases. This evolution is essential as new specified cases are continuously reinstatiated. E.g. In the past the MOUSE had only two buttons and no scroll wheel. As more and more new models of MOUSE have scroll wheels it becomes a prominent feature of the product. The cases about MOUSE containing scroll wheels have now become the common feature for the generalized case/superclass. Deviational replay is to allow new knowledge to be added to the system by experts and this knowledge is certainly absorbed to the new cases.
Model guided repairs usually apply the adaptation rules to solve new problems.

The new rules found to guide adaptation need to be retained and the new cases can be stored by referring to the capacity of the case base and the policy set by administrator.

New object classes are constructed by considering the constraints which are represented as "forbidden" in deontic logic. The sketch produced by designers does not imply that it can be exactly transformed to the final finished products since the constraints of equipment, production capacity, material cost and production cost need to be taken into consideration.
CHAPTER 6. DEVELOPMENT OF PRODUCT

INFORMATION EXCHANGE MODULE

(PIEM)

The increasing pressure of a highly competitive environment is driving corporations to develop their applications in a quick Return On Investment (ROI) manner, which urges companies to build applications that closely match their business models and processes. Industrial design companies constantly build new designing instruments such as computed-aided drawing tools; manufacturers make their job shops more flexible in order to accommodate changing customer requirements and to customize their products to fulfill the customers’ expectations. The ever-changing business environment has led to the development of Rapid Product Development (RPD) which was defined by Bullinger (2000) as an interdisciplinary evolutionary methodology to combine all influences of an engineering process into an iterative development process. Product data are constantly increasing in terms of complexity and richness. The upsurge of the World Wide Web has text, image, audio and video data that needs to be sorted and stored. Intranets and extranets drive the data workflow both within and outside the company, while forming process collaboration with partners, suppliers and customers. Manufacturing firms, for instance, use intranets to reduce the assessment time taken to qualify a potential supplier for purchase orders.

As enterprises prepare to make commercial use of the Web, all of their Information Technology (IT) applications, in addition to scalar data, must be able to integrate documents, images, sound and video clips. This would allow businesses like
manufacturing companies to store and manage all the customer data, which include product data, in a single database. As the cost of computer hardware is decreasing but the functionality has become increasingly powerful, a novel and complicated way of database applications is required to manipulate the newly emerged formats of data and in many cases distribute the complex processes across multiple nodes on a computer network such that many jobs can be processed concurrently.

Efficient access to the sorted data is necessary for decision makers. Many software application design and development teams have found that developing applications based on object technology from the ground up can best satisfy the business needs and meet the challenges of the fast changing marketplace. However, many corporate companies still use traditional databases which basically have a relational database structure. There is still no a standard to consolidate the production information within the product development environment. This hinders effective design collaboration and in turn delays the whole manufacturing schedule and time-to-market. The Product Information Exchange Module (PIEM) concerns the integration of a corporate database with a knowledge base. The product data stored in a database is converted to neutral format and assimilated to the knowledge base. Most knowledge-based systems support the neutral format but great effort is still needed to handle the migration process such as searching the relative field in the relational database and exporting to the right location in the knowledge repository. An objectization approach is suggested, which transforms a relational database schema to an object model based on object technology. Objectization turns clusters of data into the form of inter-related objects, providing relevant features such as inheritance, polymorphism and encapsulation. Once the data is restructured in the form of objects, product data can be transformed to XML format based on PIML. PIML is in the
middle tier, connecting the client-side and database. In particular, PIML requests an
organized document structure for data exchange and leverages the data information.
This concerns extracting information, making changes and querying the document.
The PIEM framework is shown in Figure 6.1.

![Diagram of the PIEM framework]

Figure 6.1 The framework of PIEM

6.1 Need for Product Information Standard

A unique data exchange standard is required and Product Information Markup
Language (PIML), which evolves from XML, is proposed to fill the gap of the
development of an expandable and flexible language for product data management.
Although most researches have conducted studies regarding data exchange using
XML, there is still lack of research to leverage the knowledge of customized product
data exchange. The World Wide Web Consortium (W3C) specification defines XML
as a subset of SGML while there are 3Cs for product development which stand for Communication, Co-ordination and Collaboration. The first C (Communication) starts with data sharing as it is essential to have a general idea what product data is stored in the relational database. According to Rezayar (2000), 50-80% of all components in products from Original Equipment Manufacturers (OEMs) are now manufactured by outside suppliers and the trend will continue in the next decade. Efficient communication between different functional departments is crucial and it is simply interpreted as delivering the right information to the right person. Right information means data that is in the right format such that the target person can read and share that information with others. It is admitted that there are increasing demands in that the computer system should achieve collaboration and communication between organizations. The internet provides not only a flexible approach to share information but also a value-added channel for co-ordaining the business process with a broader range of trading partners in order to achieve the goal of shortening the lead time and increasing agility during product development. In particular, it brings the benefits of on-time delivery and well-coordinated schedules. This implies that XML allows information integrity and the correspondents can reach a consensus decision. Collaboration requires sufficient knowledge support at different decision-making levels. Collaborative commerce facilitates the integration of parties in the supply chain. In order to achieve collaborative commerce, agile manufacturers are required to design, execute, control and adapt the manufacturing environment. Interaction is critical for managing the dynamic manufacturing activities. However, there is a great barrier between interactive communications due to the lack of a common product information standard to assimilate the large volume of information derived from a range of sources.
6.2 Features of PIML in PIEM

The proposed PIML makes it possible to exchange information across the traditional database and knowledge base by providing a seamless infrastructure for daily data entry into a relational database, and also to acquire knowledge with a knowledge toolkit. PIML is also able to serve as the product information standard to manipulate the product data by means of DTD or XML Schema. Apart from representing the XML with XML Schema in static structure format, another feature of PIML is that it represents the dynamic behavior of the objects in order to attain the goal of acquiring and mastering the latest information.

The major features of PIML are:

i. Leveraging existing web technology and grasping the future trend in data exchange formats
ii. Overcoming the barrier of an unknown interface
iii. Enhancing data structure and content checking
iv. Automatic monitoring and archiving of design changes
v. Transforming XML data to produce executable business logic

6.3 Development of PIEM

Objectization turns clusters of data into the form of inter-related objects, providing relevant features such as inheritance, polymorphism and encapsulation. Once the data is restructured in the form of objects, product data can be transformed to XML format based on PIML. PIML is in the middle tier, connecting the client-side and the database. To achieve effective communication, it is essential to solve the
problem of the sorting of enormous amounts of data in a structured way such that the
data can be achieved easily. Object technology provides flexibility for initial data
design and gives the ability to handle unforeseen events or transactions. Another
crucial advantage is resilience which is the ability of the system to handle changes
during maintenance without the occurrence of other data exchange problems for every
little change.

The object model is first transformed to PIML as a set of XML for product
data standard. This has the advantages of being able to operate in multiple platforms.
According to the following steps shown in Figure 6.2, objectization is implemented to
achieve design agility.

1. Create product development model

2. Convert it into an object model

3. Define XML schema based on object model

Figure 6.2 The major steps for objectization

6.3.1 Create Product Development Model

The dimensions of agility are summarized as enriching customer
requirements, organizing and mastering uncertainties; leveraging the impact of people
and information and co-operating to enhance competitiveness. Identifying customers’
requirements is a critical initial stage of product development. Customer requirements
can generally be identified as statutory requirements, explicit customer needs and
implicit customer wants. The explicit requirements are known as some characteristics
or features possessed by market products. That means competitors already know the
product features. The new product should include those basic features even though you know that customers will not be excited by those features. Implicit customer requirement is similar to delighters in Kano's Model (Kano, 1984). As consumer decisions are probably driven by cognition (Zaltman, 2002), it is necessary to know customers' belief and hidden requirements. Having identified the customer needs, design practitioners should find out which processes are involved in the product development cycle in order to examine the function of each process. To interconnect the process, the workflow of the product development process should be identified.

6.3.2 Convert it into an Object Model

In designing the object model, objects are identified and their relative behaviors and attributes should be realized. According to Brown (2002), attributes are the data elements carried by an entity which describes and records its state. The most typical attribute of the object is "name". The attribute is static as it does not change even though the status of the object changes. Behaviors, which specify the operations of the object, represent the dynamic side of the object. For example, "General Information", which is one of the objects in the object model, contains "Product name, Product Number, Start Date," as attributes and "Add, Delete, Modify" as behaviors. Having identified the objects within the model, system analysts can construct the object model by associating various objects.

6.3.3 Define XML Schema Based on Object Model

The object model defines a complex data structure that maps to a complex type in XML Schema. XML Schema helps define the data-type of the attributes while the behaviors can be encoded in a scriptlet. The following code shown in Figure 6.3 is
a complex type in Product Data Schema (PDS) which is the XML schema for product data.

```
<xs:complexType name="Product">
  <xs:sequence>
    <xs:element name="Name" />  
    <xs:element name="Price" />  
  </xs:sequence>
  <xs:attribute name="ProductID" />  
</xs:complexType>
```

<table>
<thead>
<tr>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>@ProductID</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Price</td>
</tr>
<tr>
<td>Add</td>
</tr>
</tbody>
</table>

Figure 6.3 PDS and UML class diagram

It is proposed that the relational database first transforms into an object model as object technology can take advantage of encapsulation, polymorphism and inheritance.

i. Encapsulation hides details and complexity within a well-defined published boundary.

ii. Polymorphism means occurring in various forms.

iii. Inheritance is the ability to associate characteristics that are common to all members of a class (type) of objects.

6.3.3.1 Association

In object oriented modeling, the relationship expressing the interaction between instances of two objects is represented by the verb that describes what objects do to each other or the roles that each object plays in its action on the others.

In this example, the object A carries out different tasks and interacts with object B prior to interacting with object C. It is found that object A plays various roles and interacts with different objects. The following XML Schema (see Figure 6.4) code
Chapter 6  Generic Methodology for Developing PIEM

illustrates the role performed by object A and the association between object A and related classes.

```xml
<xsd:complexType name="A">
  <xsd:all>
    <xsd:element name="Element1" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="Element2" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="BehaviorOfA">
      <xsd:attribute name="Attribute1" type="xsd:string"/>
      <xsd:complexType>
        <xsd:complexType>
          <xsd:element>
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element ref="B"/>
                <xsd:element ref="C"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:complexType>
      </xsd:element>
    </xsd:complexType>
  </xsd:all>
</xsd:complexType>
```

Figure 6.4 Express associations in XML Schema

An XML Schema consists of elements that are simple types or complex types. Complex types allow elements in their content and may carry child elements and attributes while simple types can contain neither child elements nor attributes. As object A has its own elements which are “Element1” and “Element2”, object A is defined as complex type. Compositors are elements that specify a sequence<xsd:sequence> and conjunctive<xsd:all> interpretation of the contents of the compositor. <xsd:sequence> is used as the elements in the group which is required to appear in the specified sequence within the containing element. According to the object modeling, object A needs to interact with object B first and then object C. In this case, the sequence to contact these two parties is essential. The conjunctive compositor <xsd:all> defines an unordered list of elements, groups, or compositors. There is a strong relationship between <xsd:all> and complex element. For those
elements appear in the containing elements (complex element), web developers can use `<xsd:all>` only when the contained elements have either `[0..1]` or `[1..1]` multiplicity (Carlson, 2001).

### 6.3.3.2 Inheritance

Inheritance is one critical characteristic of PIML. To depict what is inheritance, two terms including “subclass” and “superclass” should be clarified first. A superclass includes all the instances of the subclass, plus, possibly, some more as well. In other words, a subclass is made up of selected instances from another class, referred to as the parent class or superclass.

Inheritance is when a subclass instance, in addition to the attributes and behaviors it has, by virtue of belonging to the subclass, also has all the attributes and behaviors that instances of the superclass have (Brown, 2002). It can be expressed with the following notation in Figure 6.5.

![Figure 6.5 Generalization and specification](image)

---

6.10
Chapter 6  Generic Methodology for Developing PIEM

Generalization and specification, which is another approach to explain the
relation of the superclass and subclasses, is used for the inheritance subclass hierarchy.
Superclass generalizes its subclasses by including all the features that are general to
all the subclasses whereas subclass specializes its superclass by possessing some of
the superclass instances. The common attributes shared by subclasses become the
attribute of the superclass.

In Figure 6.6, subclass $C_1$ and $C_2$ are shown as descendents from superclass
$P$. Those subclasses inherit the attributes $(a_1,a_2,a_3)$ from the superclass. Apart from the
attributes and behaviors from the superclass, the subclass possesses its own specific
attributes and behaviors. The following XML Schema shows inheritance subclass
hierarchy in PIML.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by Carman (HKPU) -->
<schema targetNamespace="http://www.altova.com/IPO"
elementFormDefault="qualified" attributeFormDefault="qualified">
<complexType name="P">
  <attribute name="a_1"/>
  <attribute name="a_2"/>
  <attribute name="a_3"/>
</complexType>
<complexType name="C_1">
  <complexContent>
    <extension base="iio:P">
      <attribute name="a_4"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="C_2">
  <complexContent>
    <extension base="iio:P">
      <attribute name="a_5"/>
    </extension>
  </complexContent>
</complexType>
</schema>
```

Figure 6.6 Express inheritance with XML Schema
Polymorphism can be interpreted as many (poly) forms (morphism). Polymorphism, which is one essential concept of the object-oriented paradigm, can be transformed and applied in product development. Polymorphism can be generally classified into two different grouped ad-hoc polymorphism or uniform polymorphism. Overloading and coercion are under ad-hoc polymorphism where parametric, inclusion and invocation are classified into uniform polymorphism.

Overloading polymorphism occurs when the same name is used to denote several different definitions. An overload procedure can be defined as \( f: P \rightarrow Q \) where \( P \) and \( Q \) are two finite sets of types with \( P \) having at least two members. The procedure (create) is applicable to any design drawing contained in the Design Drawing Start Form. There is usually a different code for each different design drawing. When \( P(\text{design drawing}) = \{ \text{concept drawing}, \text{presentation drawing}, \text{engineering drawing} \} \), there are three pieces of code to implement which activates different kinds of graphic software to be opened for the specific drawing.

Coercion is used to reduce type error by coercing one data type into another related data type. E.g., it is expected “datetime” should be input into the starting date of the form but the user inputs the “date” value into that field by mistake. By coercing the date value into a datetime value a type error situation is avoided. Since the coercion is automatic, a parameter can accept more than just “datetime” type arguments and it can accept any type that can be coerced to a “date” type.

Parametric polymorphism occurs when there is a single procedure definition and consequently a single piece of code that is used uniformly regardless of the types of arguments. A parametric polymorphic procedure \( f \) can be defined as \( f: P \rightarrow Q \) where \( P \) and \( Q \) are two sets of types, both potentially infinite. Procedure \( f \) is
applicable to values of any type \( B \) in \( P \) uniformly. For example, the cost function in injection moulding can be defined as cost:\([\varnothing] \rightarrow \text{float}\) where \( \varnothing \) is a type variable which can be instantiated as any concrete type. The cost function applies to a list of products produced by injection moulding with any plastic product and executes the same piece of code for all on such a list.

Inclusion polymorphism is when an object belongs to more than one type, specifically where one type may be a subtype of another type. An inclusion polymorphic procedure \( f \) (change the car body from block to streamline) which applied to a superclass (car), can also applied to subclass (remote car) where a remote control car is a subclass of car. In other words, inclusion polymorphism is an inheritance method.

Invocation polymorphism is not similar to other polymorphism. It is an abstraction of the definition of function \( f \). Argument \( L \) is a polymorphism argument of parameter with a list of \( 8 \) where argument \( M \) corresponds to the parameter \( W \) with a list of \( 4 \). \( WAverage \) is applicable to argument \( L \) and \( M \) by subdividing argument \( L \) into two application regions and leaving argument \( M \) with only one application region.

The application of \( WAverage \) to \( L \) and Argument \( M \) is a polymorphic invocation. To illustrate invocation polymorphism in production, product \( L \) needs to go through the following sequence (Mill \( \rightarrow \) Lathe \( \rightarrow \) Grind \( \rightarrow \) Drill \( \rightarrow \) Mill \( \rightarrow \) Lathe \( \rightarrow \) Grind \( \rightarrow \) Drill). Product \( M \) needs to go through the following sequence (Mill \( \rightarrow \) Lathe \( \rightarrow \) Grind \( \rightarrow \) Drill). The average production time can be applicable to product \( L \) into two application regions and leaving product \( M \) with only one application region. The application of average production to product \( L \) and product \( M \) is a polymorphic invocation.
6.4 Applications of PIEM

The ability to access and exchange data between various parties such as vendors, customers and manufacturers is an issue that needs to be addressed. Breaking down the barriers among the parties is beneficial to the collaboration between partnerships and ensures a smooth information flow among business partners. Once the data in the relational database has been objectized, the data is structured in an object model. This paves the way for deploying the PIML. PIML consists of three major components: XML, Extensible Stylesheet Language Transformations (XSLT) and a script language. These provide a universal product information exchange standard to enable the knowledge workers to acquire sufficient and accurate information. In particular, PIML is mainly applied to product design, encompassing information about manufacturing processes and engineering analysis.

PIML facilitates the transformation of data into 3 different formats such as:

(i) XML for other systems--- PIML can be transformed into several outputs to various audiences by applying different XSLT documents where XSLT is a language for transforming XML documents and XML Path Language (XPath) which is an expression language used by XSLT to access or refer to parts of an XML document. PIML reads the XML source document and the associated XSLT style sheets. Once PIML parses XML files, and their associated XSLT files, into trees of a node, each node corresponds to an XML document element or attribute. An XSLT transformation is applied to the source trees to produce result trees according to the specification of the XSLT. The result trees are serialized as output files which are in XML, HTML, or other text formats.
(ii) Text format for knowledge based systems—XSLT can also transform the data in text format which can be exported to a knowledge based system since the data in neutral format can act as the data source for knowledge based systems. The updated data in a relational database, which has been objectized and transformed as XML, in turn becomes a valuable element of the knowledge repository. Users can leverage the knowledge which is manipulated by the inference engine through an user-interface. The inference engine provides the reasoning capability and provides advice for various problems. In this research, inferences occur in CBR when past similar cases are adapted for product development activities. The simple logic flow of decision-making can be encoded with XSLT, which is a programming language with flow control and contains the following tag such as `<xsl:if>`, `<xsl:for-each>`, `<xsl:choose>`. Those tags can assist design practitioners to extract explicit knowledge. In order to achieve interaction and run the complex logic flow of decision-making, XSLT allows vbscript, java script, C++ to operate with the XML. The dynamic behavior of objects is performed by loading java script and vbscript.

(iii) HTML for the suppliers and customers—The authorized parties can view the data in a more familiar format (HTML) since HTML is more presentable than XML as it holds Meta data annotated by tags.

6.4.1 Data Transformation to a Knowledge Base

The Internet appears to have latest technology and it is beneficial to globalize electronic information sharing. XML is recognized as a universal standard for the exchange of transaction documents over the Internet. PIML is a specific set of
XML for product development which takes advantage of the well-structured XML schema to provide a well-understood syntax and self-defined markup language to suit particular needs in terms of product data exchange.

6.4.1.1 Product Data Schema (PDS)

During the product development life cycle, common changes include changes to design method, changes to hardware, the provision of better quality data, changes to the organization of manufacturing system and changes to planning and control methods. The purposes for engineering changes are error correction and continuous improvement. Engineering changes need strong collaboration among change requestors, change brokers and change providers.

i. Change requestors, who initiate the change, can be the clients or technicians in engineering departments.

ii. Change brokers are co-ordinators for the impact activities and operations for each affected object. Change requestors can interrogate change brokers in order to collaborate with required change providers.

iii. Change providers effect the change to products, components, assemblies, process, document or even the activities in the supply chain.

The PDS allows users to define the structure and data type for XML documents. The change types of engineering changes can be generally categorized as adaptive, corrective, perfective and preventive (OMG, 1998). Table 6.1 shows the major elements in engineering change requests.
### Table 6.1: Major elements of engineering change request

<table>
<thead>
<tr>
<th>Object Name</th>
<th>Element</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EngineeringChangeRequest</td>
<td>SourceItem</td>
<td>string</td>
<td>Item that need to be replaced</td>
</tr>
<tr>
<td></td>
<td>TargetItem</td>
<td>string</td>
<td>Item that need to replace</td>
</tr>
<tr>
<td></td>
<td>BOM</td>
<td>string</td>
<td>BOM which item belong to</td>
</tr>
<tr>
<td></td>
<td>EffectiveDate</td>
<td>date</td>
<td>The date when the ECR was activate</td>
</tr>
<tr>
<td></td>
<td>ApprovalStatus</td>
<td>string</td>
<td>Type of approval: Approval or decline</td>
</tr>
<tr>
<td></td>
<td>ChangeType</td>
<td>string</td>
<td>Type of change. The choice includes: adaptive, corrective, perfective and preventive</td>
</tr>
<tr>
<td></td>
<td>Problem</td>
<td>string</td>
<td>Problem leads to design change</td>
</tr>
<tr>
<td></td>
<td>Solution</td>
<td>string</td>
<td>The proposed solution for the problem</td>
</tr>
</tbody>
</table>

PIML describes the specification by XML where PDS validates the incoming and outgoing XML document. Figure 6.7 shows PDS which expresses some typical element in the engineering change request based on Table 6.1.
Figure 6.7 PDS of engineering change request

6.4.1.2 Mapping Mechanism between a Relational Database and a Knowledge Repository

In PIIML, close relationship between the knowledge repository and the database is established as knowledge workers operate and conduct activities with documents and data which are stored in the database. The typical tables in a database are categorized as customer, order, order details, employee and supplier, where the case stored in the case base may include relevant pieces of information from various tables in the database in addition to explicit knowledge. To interconnect the heterogeneous repository, XSLT is deployed to manage semantics of XML document. XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary (W3C, 1999). Almost all commercial DBMS such as Oracle 8i, IBM DB2,
and Microsoft SQL Server have been extended to handle XML documents. The
XSLT processor can convert the XML document to the specified format such as text,
html or xml according to the template rules. The output data can be viewed in a web
browser or act as a data source for a knowledge based system.

XSLT possesses programming logic since XSLT supports numerous sets of
flexible data types, a full set of operations and programming flow logic. The
following example is to illustrate how XML combines with XSLT which captures
programming logic on PIML. Supposing the engineering change request contains the
data such as source item, target item, BOM, effective date, approval status, change
type and change request. Data stored in the relational database is exported as the
following XML format (See Figure 6.8).

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by carman (HKPU) -->
<?xml-stylesheet type="text/xsl" href="C:\My Documents\Conf\ProglECR.xsl"?>
<EngineeringChangeRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\My Documents\Conf\ProglECR.xsd">
  <ECR ChangeType="Preventive">
    <SourceItem>ABSCarBody011</SourceItem>
    <TargetItem>ABSCarBody012</TargetItem>
    <BOM>ModelCar100</BOM>
    <EffectiveDate>2003-03-01</EffectiveDate>
    <ApprovalStatus>Approved</ApprovalStatus>
    <ChangeType>Corrective</ChangeType>
    <ChangeRequest>For silver bath, Condition: Temperature is 24-32C, Current density is 0.5-1.5A/dm sq, Agitation is preferred. Cathode efficiency is 100%, Ration anode to cathode area is 1:1, Anodes is silver</ChangeRequest>
    <Reason>Maintain the tank condition.</Reason>
  </ECR>
</EngineeringChangeRequest>
```

Figure 6.8 XML of engineering change request
Perfective engineering change requests (Highlighted by the dotted line in Figure 6.8) may contain critical information for product design in the plating process. Descriptions of engineering changes are stored as case solutions in the case base. E.g., design practitioners need to recognize the function of the designed parts, the chemical and physical properties of the plating design such as strength, hardness, and brightness. As they identify how plating processes improve the properties, they need to analyze the cost and benefit for the defined surface finish. Some geometrical design principles of the part need to be taken into consideration. Sharp edges, ridges, recess bind holes etc introduce difficulties and increase the cost of the finishing operation. In the XML code shown above, the design department requests a design change due to the sharp edges and the right angle. Design practitioners also state the reason for engineering change request. That information in the database is exported in XML format and then imported to the case base. The text file shown in Figure 6.9 is imported into the case base and acts as a knowledge source for the CBR system.

```
CaseName: Design for Plating
Description: Part receives very little plate in an acute angle. Every sharp protruding edge draws extra current and builds up with extra plate and vice versa. All sharp edges and angles should be rounded to the greatest degree design allows.
Solution: Consider not to include sharp edges and right angles for product design.
```

Figure 6.9 The neutral data file for the case base
The data incorporated can be migrated to the knowledge repository of CBR.

The knowledge representation relevant to CBR can assist junior process or product engineers to formulate a possible solution for a particular domain.

6.4.2 Web-Based Support for Product Data Management

PIML requests an organized document structure for data exchange and leverages the data information which deals with extracting information, making changes and querying the document from code. To meet this objective, the Document Object Model (DOM), which has achieved remarkable attention and deployment in recent years, acts as a key component of the XML family of standards (Suzuki and Yamamoto, 1999). W3C defined DOM as a standard Application Programming Interface (API) to the document structure while DOM aims to make it easier for programmers to access components for deleting, adding or editing their content, attributes and style (W3C\(^1\), 2001). Only programming with script brings out the dynamic status or functionality of the user agent. This leads DOM to stand in the context of scripting and facilitate XML application. DOM contains five major components which are DOM Core, DOM HTML, DOM XML, DOM Event, DOM CSS(Cascading Style Sheet) (W3C\(^1\), 2001; Ceponus, 1999). DOM Core relies on the tree-like hierarchy representation of the document where node, root and sub-node describe the element in XML(Wilson and Wilson, 2000). DOM HTML, DOM XML, DOM Event, DOM CSS(Cascading Style Sheet) provide a platform or method to manipulate HTML, XML, event and CSS respectively. Figure 6.10 shows the electronic form of engineering change request generated by XML and XSLT.
This chapter proposes PIEM that facilitates data exchange at different design stages in distributed manufacturing environments for rapid product development. The conceptual idea of the proposed modules has been presented and the implementation steps will be discussed in the next chapter.
CHAPTER 7. IMPLEMENTATION OF RPDS

Chapter 7 focuses on the implementation of RPDS including four chronological stages which are project startup stage, structural formulation stage, testing and evaluation stage and finally the operational stage. As the system development lifecycle in this research is based on the roadmap shown in Table 7.1, the analysis, design, programming and system-testing activities are four stages. It should be noted that these activities are, particularly on a medium sized project, to be divided into separate stages themselves in line with the boxes in Table 7.1.

Table 7.1 The roadmap for implementing RPDS

<table>
<thead>
<tr>
<th></th>
<th>Startup Stage</th>
<th>Structural formulation Stage</th>
<th>Evaluation and Testing stage</th>
<th>Operational Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Based Product Development Module</td>
<td>Establish a timeline with milestones, contingencies and contributions</td>
<td>Analyze and design system with object technology</td>
<td>Validate with OO design metrics</td>
<td>Keep track of IS adaptation and evolvable objects</td>
</tr>
<tr>
<td>Hybrid Knowledge Module</td>
<td>of each participant and prepare the network infrastructure and system environment</td>
<td>Design and build case base; reuse the knowledge by adaptation</td>
<td>Find out the competency of cases in CBR</td>
<td>Carry out case base maintenance</td>
</tr>
<tr>
<td>Product Information Exchange Module</td>
<td>Setup data structure and XML semantics</td>
<td>Validate XML with XSD</td>
<td>Carry out business transaction</td>
<td></td>
</tr>
</tbody>
</table>
It is instructive to consider the systematic procedure in various stages of OBPDM, HKM and PIEM in RPDS. Table 7.1 shows more explicitly how each module goes from the initial stage to the final stage in the horizontal direction, and the integration of three modules at each stage in the vertical direction.

7.1 Project Start Up Stage

This stage is sometimes called the initiation stage and covers the work which is carried out at the beginning of the project when the basic framework is put in place. Most project owners would like to be right the first time. Although this seems to be too ideal, getting things right the first time can be achieved if the right person does the right thing at the right time. The project start up stage is an important stage of the project as it is where the foundations are laid. Much of the future work of the project is based upon these foundations and the importance of this stage should not be under-estimated. If the initial step is not carried out properly, the failure rate of the project will be high or much effort is required for reworking it. The recovery situation may not be as good as the situation of “right at first time”. It is necessary to consider the resource allocation by asking “what”, “when” and “who”. While considering the methodology and reasons, the question about “how” and “why” need to be raised also.

7.1.1 Technical Level

The manufacturers in PRD in 90’s regarded it is risky to put more investment on high technology and they thought they could continue their business with low tech-end product provided that labor cost in PRC was low. However, customer driven requirements do not allow manufacturers to stick to low technology.
High technological products are usually fabricated by complicated processes and an enormous amount of information is embedded behind the complex process. A database is used to store the product data and the value of process parameters setting. Work flow, material flow and information work are all inter-related and pass from the concrete physical layer to an abstract information layer. Object-oriented modeling, which helps to identify the association and interaction of objects, can be found in material flow and information flow that can be represented as business modeling and software modeling respectively.

Apart from software, hardware and network infrastructure play important roles in implementing RPDS. Hardware includes switches, hubs, routers and computers acting as sever and clients whose configuration should be determined according to the software requirements. The essential software includes Microsoft® SQL Server 2000, CBR (Case Advisor4.02), and browser (which include XML parser). Maintaining a good computer environment includes maintaining the network infrastructure (intranet and extranet) and computer storage environment (whether the database and server kept by Application Service Provider (ASP) or by company itself) that streamlines the implementation process. Some elements listed in Table 7.2 need to be considered. The technology level does not concern only the hardware and software, it is necessary to consider who controls the technology and whether the technological level can fulfill the needs of the customers.
Table 7.2 The technical aspect of implementing RPDS

<table>
<thead>
<tr>
<th>Network Infrastructure</th>
<th>Consideration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intranet</td>
<td>Knowledge sharing portal</td>
</tr>
<tr>
<td></td>
<td>Basic communication such as e-mail</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Availability of network in PRC plant site</td>
</tr>
<tr>
<td></td>
<td>Transfer spend and performance of the network in PRC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System environment</th>
<th>Consideration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provided by ASP</td>
<td>Application expertise</td>
</tr>
<tr>
<td></td>
<td>Services expertise-security</td>
</tr>
<tr>
<td></td>
<td>Availability, reliability of service providers and make sure the service provider fits into company’s five year plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maintained by company itself</th>
<th>Consideration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Contingency plan</td>
</tr>
<tr>
<td></td>
<td>Maintenance plan</td>
</tr>
<tr>
<td></td>
<td>Database Backup plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General</th>
<th>Consideration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability of power cable (esp. in PRC), availability of power supply to equipment, humidity, temperature</td>
<td></td>
</tr>
</tbody>
</table>

### 7.1.2 Organizational Level

Mastering the technology is far more important to owning the high technological facilities. The technology cannot be enhanced by purchasing hardware or software. Human expertise is required to help in the execution to apply the technology and produce value-added products and services to enhance company’s competitive. This vision is to establish the basis of a framework of how RPDS can be implemented or tested. It is generally classified as top-down versus bottom-up. Top-down begins from the high-level perspective and disseminates to the deeper level of detail development while bottom-up starts from performing the key activities at the lowest-level to achieve the business process goals in order to build up a high level strategy. It is important that the customer side (users) is involved in the project planning as many projects fail, not through technical resource problems, but through lack of available user resources at appropriate points in the project. To clearly define the roles and responsibilities is the crucial step at the start up stage. Table 7.3 has summarized the role and responsibility of those involved.
Table 7.3 Roles and responsibilities

<table>
<thead>
<tr>
<th>Roles</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project director</strong></td>
<td>He/She is responsible for the overall project which involves co-ordination of the user resources and all issues relating to the user and software provider. Arrange for the appropriate user involvement, specifically during the production of the requirements specification and for the acceptance of the system.</td>
</tr>
<tr>
<td><strong>Stage manager</strong></td>
<td>Responsible for delivery after each stage</td>
</tr>
<tr>
<td><strong>Project manager</strong></td>
<td>Demarcation lines are clearly agreed and documented at the start of the project as misunderstandings and problems over responsibilities can arise later in the project if the roles are not clearly defined.</td>
</tr>
<tr>
<td><strong>User</strong></td>
<td>Prototyping, training and production of user manuals are some of the other activities which require user input.</td>
</tr>
<tr>
<td><strong>Trainer</strong></td>
<td>Provide training for the user. Find out the difficulties faced by users.</td>
</tr>
</tbody>
</table>

Identifying obstacles to success throughout the project; analyzing the impact of the issues and assigning individuals or teams to resolve them are three major activities in the start up stage. Resource allocation is critical for project implementation where resource includes human resources, time, capital and equipment. As each person’s contribution is necessary to ensure success, either from within the company or from outside resources, each person should prepare a complete skills assessment based on the major phases of the initiative. To keep track of the progress of the project, a timeline with milestones, contingencies and contributions of team participants are needed.
7.2 Structural Formulation Stage

In the structural formulation stage, most of the design and analysis work is carried out to transform process design to system architecture. OBPDM is constructed by studying the objects in the problem domain and the solution domain. The establishment of HKM includes acquiring case features and representative cases by considering the case distribution. Data migration allows data replicated from the database to a case base. The methodology of data migration needs to be formulated in this stage. PDEM is set up by introducing a synonym of “ontology” and universal data exchange format that acts as a bridge between the database and the knowledge base that were described by the pre-defined standard.

7.2.1 OBPDM--The Requirement Analysis for Object Oriented Application and Object-Oriented Design

Object technology is becoming widespread in system development. It is the core technology used in OBPDM. Object-oriented software development includes object oriented analysis and design as well as object oriented programming language. Object-oriented analysis and design is adopted in the structural formulation stage. Object-Oriented Analysis (OOA) is to analyze the real world by studying the business processes or activities and depicting them as a list of object-oriented requirements. OOA is a paradigm shift between the functional shift between the functional requirements phase and the object-oriented development phase. Object-Oriented Design (OOD) is used to model the real world to a higher level of abstraction in order to deal with complexity by generating a greater stability of objects. Object technology can provide advantages of easier maintenance for
changes or new features. "Use case" makes system analyst identify the problem
domain and the operations and interaction of some actors under the specified context.
Steps of OOA and OOD in the structural formation stage includes (I) scripting the
procedure for the actor (II) realizing the structure of the objects/classes
(III) specifying the collaboration among objects (IV) specifying the state of each
object.

(I) Scripting the procedure for the actor

According to the functional requirement requested by users and considered
by the system analyst, a sequence of steps is performed by users and a response is
given by the system. The human-computer interaction is depicted by finding the
communication association of each actor. The used case can be generalized by
finding the common factors of used cases and including dependency to reuse some
common behaviors performed by actors in used cases. TRMI, which can be
instantiated and perform individual operation under specific actors, provides agility
for enterprises to cope with unexpected events. TRMI is especially suitable for
turbulent markets.

(II) Realizing the structure of the objects/classes

The objects found in product development processes from the real world
can be used and modeled with object modeling technique such as UML. The
relationships with other objects can be depicted after the classes, attributes and
behaviors of the object have been identified. The structure can be formulated by
defining the associations between objects one that is a generalization-specification
structure or a whole-part structure.
(III) Specifying the collaboration among objects

As when studying the scenario of business operations or the manufacturing environment, it is not difficult to find that objects that make up a system interact and collaborate with one another to provide the functionality of the system, thereby solving a particular problem. To achieve collaboration, good communication is essential. Efficient communications allow a sender object to send messages to a receiver object element to convey information and request processes where the operation is regarded as "trigger an event". The stimulus received by one object may trigger another event and affect other objects. A chain of interaction may occur in sequence and each element involved may have its own life span as is indicated in the proposed sequence diagram (Figure 4.2)

(IV) Specifying the state of each object

An object can survive in the form of various states such as inactive, active and a suspended state. These indicate the condition of an object. An object undergoes transition when an object changes from one state to another state. The input and output of data may be different since action is performed and events occur on the way from the source state to the target state.

The major problem found in OOA is that requirements are usually imprecise or incomplete as information may overlap or be inadvertently missing. Class diagrams, sequence diagrams and state diagrams help to avoid duplication by linking similar objects or classes together and by making use of the object-oriented characteristics such as inheritance, polymorphism and encapsulation.
7.2.2 HKM---Building the CBR System

HKM is characterized by its ability to handle two main types of data from dissimilar database system including relational database and knowledge base in order to support knowledge discovery.

7.2.2.1 Defining the Case Structure

A case is composed of problem, index and solution. The description of a problem describes the symptoms of the problems and the solution shows the methodology or some diagnostic steps rather than the reason for resolving the problem. Case index is designed for efficient retrieval of a particular case from among numerous cases in the case base. If users want to find the case, the search function can be used to locate the relevant case number quickly by virtue of the well-established capability of relational database tools on the market such as Microsoft® SQL Server™.

7.2.2.2 Authorizing the Case

The case can be constructed using two approaches which are: by importing one from the database or by domain knowledge workers creating a new case. If the information that composes the case can be found in a database, the case can be authored by integrating the scattered cases from various tables. On the other hand, if there is not enough information to form the case, experts must be invited to advice on how to solve the problem. The knowledge audit helps to acquire cases to build the case base.

i. Importing from a database---To enable data communication to take place between the relational database and the case base, a special data table is
created with data fields including case index, case name, case description, case and keywords. As the relational database management tool is characterized by its efficient searching mechanisms for data stored in the data fields of a data table, the data in these fields can be managed efficiently. With this table, users can locate the query case quickly for further processing.

To effectively integrate the operations of the relational database and knowledge repository, the data field ‘Open Relevant Case File’ contains a macro routine (a sequence of commands of operations), which enables the calling up of the associated word processor and loads the code (normally in text form) of the relevant case into the word processor program. This enables users to view or update the content of the case. Most commercial relational database tools have the capability to create this macro function. With the direct connection to the CBR tool, the unique reasoning mechanism of the CBR tool can be used for specific purposes. In the CBR tool, the case base stores cases of various natures just like separate data files albeit in the free text format due to the large variance of data for various cases. This proposed technique takes advantage of the well-developed data fields search engine of a relational database tool for handling the large collection of cases housed by the case repository, the storage technique of which depends on various CBR tools.

ii. Creating a new case---The new representative case can be acquired by referring to past documentation or technical reports or by interviewing the experts in a particular knowledge domain. It is quite difficult to construct the case base in an organization that owns tacit knowledge only in that particular knowledge domain. That means most knowledge, which is implied
knowledge stored in forms of subject adjustment, experience and cognitive values, needs to be retrieved by a knowledge audit. A knowledge audit helps to map the knowledge flow and transfer that turns tacit knowledge into explicit knowledge. The detail of a knowledge audit is not within the scope of this research. Such details can be studied by from the relevant references (Mcdermott, 2004; Hamilton, 2002; Tiwana, 2000). As cases may be acquired by being imported from a database or by being created in different time phases, it may lead to a heterogeneous case base where cases have various record structures. An effective retrieval method is therefore necessary to cope with such a heterogeneous case base.

7.2.2.3 Study of the Retrieval Mechanism and Setting Questions for Cases

The questions should be set according to the reasoning procedure. For example, users find pits in nickel sulfamate plating line. The system may ask several questions to find out the solution for the users. The system may ask whether there is enough anti-pitting agent in the bath. If the bath does not contain enough anti-pitting agents, bubbles tend to stick to the plated surface and pits are formed. The system may suggest that engineers monitor the level of anti-pitting again by measuring the surface tension of the bath, which should be 30-35 dynes/cm, on a regular basis. If users find that it is not caused by lack of anti-pitting agent, the system will continue to ask whether there are organic impurities in the bath. The system can understand and gain more information about the problem through asking users a series of questions. The aim of setting questions related to cases is to guide the users to find out the solution for the query case. The numbers of questions to be answered by users is based on the position of the query question and the destination of the target case in the induction
tree where the space state search is performed. As a result, retrieval time varies with
the number of nodes in the query tree. Assigning an adequate index can improve the
retrieval efficiency especially if the case base is large.

7.2.2.4 Reusing the Knowledge by Derivational Adaptation

Since the concept of specification and generalization is adopted in case
adaptation, a common slot should be identified and the rule or principle can be apply
to derive another new case that provides a specific solution for the query. The rules
are usually extracted by interviewing the experts or are based on the
handbook/standard or current practices. On the other hand, rules can be derived by
comparing the content (problem, solution and index) of cases under the same category
and by finding out the reason that leads to various solutions under different queries. If
the forward chaining rule is applied in derivational adaptation, constraints can be set
by finding what is permitted, forbidden and obligated in order to find out the best
solution for the users.

7.2.3 PIEM---Semantics of Product Information Markup Language
(PIML)

In PIEM, a product data exchange standard, the PIML, which is composed
of a set of well defined Product Data Schema, is introduced to provide a flexible way
to encode structured data into a format. In order to respond to the rapid changes in the
production environment or market trend, PIML interfaces with CBR in such a way
that it can provide advice for the design engineers to resolve the engineering problem
and minimize the ripple effect caused by sudden changes. The intelligent operation is
an artificial intelligence problem solving technique, CBR, which retrieves the past
most similar case by weighting various problem parameters and then it adapts the past solution to the new problem.

(I) Defining a glossary of product development activities

The glossary is a list of technical terms in some specialized field of knowledge which may be found in daily product development activities. Various terms may be used to describe the same object. For example, BOM can sometimes be called product structure or product architecture. As a result, a unified term should be defined to describe the particular object. BOM contains the product description, the quantity required, the name of the component, and the source of each part (purchased or manufactured) and where the information can be found in a database.

(II) Defining the data type of product data

PIML is a kind of XML vocabulary which acts as a XML schema for a product data model that allows data exchange between customers and suppliers and the enterprise. System analysts need to define the major elements, attributes, data types and constraints on a complex content definition.

(III) Constructing PDS for validation

PDS provides the function of inheritance that enables knowledge reuse. XML parses the incoming XML data files to ensure that the data from the database conforms to the pre-defined schema in XSD format as knowledge representation. PDS helps define the data-type of the attributes while the behaviors can be encoded in a scriptlet. An XML Schema file defines the format and the data types for the XML data files.

(IV) Pictorial Representation of PIML

A generic product development process consists of several processes which are: needs recognition, design specification, conceptual design, detail design,
production and marketing. As data is loosely organized and scattered during the product design stage, PIML defines an XML encoding scheme that enables a total guideline to be formulated for the product design process. The major phases and processes of each phase are shown in the hierarchy tree in Figure 7.1.

![Diagram of product development process]

Figure 7.1 Generic workflow of product development

### 7.3 Evaluation and Testing Stage

In the testing and evaluation stage, some evaluation criteria that analyze and evaluate the system performance are proposed. High quality software can be obtained through testing where high quality software means the software gives the user a correct output after he inputs the variable that has the right data type and which is
within the pre-defined boundary. According to the features of RPDS which are agility, self-learning ability and customizability, the evaluation criteria are set for verifying the design requirements. The performance can then be measured by the quantitative results of the test.

7.3.1 OBPDM--- Test for Agility

The agility of the OBPDM can be evaluated by the criteria of responsiveness to change and flexibility of the system to the changing environment. Dove et al (1996) identify the importance of an adequate management of information system to support agility and introduce the following elements to assess the agility for manufacturers which are (1)Creation, (2)Augmentation, (3)Comparison, (4)Migration, (5)Modification, (6)Correction (7)Variation, (8)Expansion and (9)Reconfiguration.

The hypothesis of quality assurance of OBPDM is established and corresponds to the elements suggested by Dove et al.

H1 (Creation): The number of functions based on the business needs in the inception phase with the evaluation formulas of the object-oriented functional models determines the development of the system.

H2 (Augmentation): refers to the viability to add or remove the classes from the existing object model without affecting the logic and the flow of the system.

H3 (Comparison): Comparability between object models is determined by counting the association and classes used when constructing a conceptual schema.

H4 (Migration): The migration ability of the legacy system to new system is determined by the number of new techniques for the system. New technology and newly-designed semantics require more effort is to migrate the model when a change of requirement is made.
H5 (Modification): The number of modified classes in one system development cycle can be useful to determine whether the solution tends to fit the specified model in an efficient way. Discrete and evolving objects have a higher tendency for modification and simple object models make modification easier.

H6 (Correction): The number of corrections can act as an indicator of construction inadequacy and language inadequacy. The rate of bugs and errors found can be used to measure the correctness of the designed object models.

H7 (Variation): Variation can be achieved by adopting polymorphism as polymorphism overrides the inherited operation for catering for a specific situation. A high rate of polymorphism leads to high variation which may have a negative impact on the clarity property of a model.

H8 (Expansion): Inheritance is a way to expand with object technology. Classes with numbers of children are very difficult to modify and correct if errors are found.

H9 (Reconfiguration): Reconfiguration of a model is to support one particular element (class, object or even a method) and move it from one system unit to another unit in order to provide different solutions. The reconfiguration can be measured by the frequency that an element is used in other system unit.

The number of object-oriented metrics is employed for the structural and dynamic modeling for OBPDM. Structural modeling metrics includes number of attributes, number of subclasses to each superclass, depth of inheritance, number of clusters, and number of visible classes. Dynamic modeling metrics includes number of states in a class, number of interactions between classes, number of classes overridden and number of methods invoked by message.

The complex product data is encapsulated in objects which encode the product development process. The object model makes use of inheritance and
polymorphism which are the characteristics of object technology. As a result, object-oriented testing rather than general software testing should be adopted for object-oriented application. The procedure of object-oriented testing is listed as following.

i. Identify the scope of the test

ii. Identify what kind of characteristics of object technology needed to be test

iii. Realize the reason for the test

iv. Find out what test is to be used and the testing procedure

v. Perform testing

vi. Analyze the result

vii. Document the errors found in the test

The data collected in testing can help to validate whether the hypothesis is correct or not. Applying the statistical analysis such as correlation analysis or parameter analysis can help to prove the relationship of the metrics involved in the hypothesis. It is found that OBPDM is advanced in variation, expansion and reconfiguration by employing TRMI to achieve the goal of responsiveness to the external stimulus by reconfiguring the various methods to accomplish the same operation.

7.3.2 HKM---Test for Self-learning

The learning ability of the HKM can be evaluated by the adaptability test, performed by CBR. Equipped with a robust and cost effective suite of supporting capabilities, the CBR system under HKM also allows the design team to work in the window environment with which it is most familiar and the CBR system provides up-to-date solution for the designers to design an innovative product. The same
Chapter 7. Implementation of RPDS

evaluation approach for OBPDM is adopted where hypotheses are generated and
tested by defined metrics. The confidence solution given by the CBR system is based
on the similarity scores for the retrieved case, the typicality of the subject with respect
to the case base, the deviation of solutions and the percentage of correct solutions
suggested by the retrieved case (Cheetham, 2000). Those factors mentioned before
stress the confidence in retrieved case but ignore the confidence in adapted cases.
Since the main feature of RPDS, learning ability, is performed by HKM, more effort
needs to be put into the evaluation of the adapted cases. In the CBR cycle, cases are
first retrieved prior to being adapted and reused to provide a new solution for the
query case. The confidence measurement of adapted cases is aggregated with the
confidence measurement of retrieved cases. As a result, in addition to retrieval
accuracy, consistency and uniqueness, adaptation accuracy and consistency ensure
that a practical solution is proposed after case adaptation.

H1 (Adaptation Consistency): If the same source case is retrieved after the second
same search, this is regarded as retrieval consistency. The consistency of case can be
measured by the deviation of the adapted case. Tests can be performed to check that
the values of attributes are within a reasonable range.

H2 (Adaptation Accuracy): Errors can be a counter indicator of accuracy. In general,
the fewer rules used in case adaptation, the higher the accuracy of the adapted cases.
The similarity score of the case paves the way for accurate adaptation since a high
similarity score means that it has high probability of being applicable in reality.

H3 (Solution Uniqueness): Uniqueness means there is no alternative case that exists
to solve the same query in exactly the same way. Since the proposed methodology
about HKM is to subsume similar cases under one generalized case and perform value
adjustment when query is made, uniqueness is found in the adapted case rather than in
the case stored in the case base. The uniqueness of the solution can be measured by whether the adapted case can suggest different solutions when different queries are made.

Having introduced the metrics for checking the case adequacy, consistency and uniqueness, the procedure of CBR testing is listed as follows:

i. Obtain a number of representative cases

ii. Evaluate each hypothesis with the corresponding metrics

iii. Apart from getting statistic result from the experimental testing, collect expert comments by asking users to fill in the evaluation form.

iv. Analyze the result

v. Refine the case and improve the adaptation mechanism if the acceptance level of accuracy, consistency and uniqueness cannot be met

It is found that the adaptation consistency can be maintained initially. However, as the case base is continuously updated by the data from the relational database, the adaptation consistency cannot be maintained because more new cases are created. The cases may evolve as new data continues to update the case base from the database. It is found that having well-designed rules and firing the right rule is more important than the number of rules fired. Normally, the fewer rules used in case adaptation, the higher the accuracy of the adapted cases. In general, the unique retrieved cases may generate unique adapted cases except that if some special adapted rules are imposed to make the adapted cases coherent then the uniqueness of the adapted case may be lost.
7.3.3 PIEM --- Test for Customizability

The customizability of the PIEM can be evaluated if the XML is tailor-made to the users’ needs. As XML can be self-defined to fit personal needs, a consistent XML document from the semantics point of view is needed. In the research, PDS is used for PIML validation and intelligent editing. XML Schema takes advantages over DTD as it enables users to create their user-defined data type and provides better relationships between elements. On the other hand, DTD has limited data types and does not support namespaces. Figure 7.2 shows a schema overview of the XSD file. It can be seen that the parts with the attributes namely, description, partID and type are shown in the lower part. Product family, which is the sub-element of the parts division, can be sub-divided into process, material and appearance. Some parts, which may undergo similar sequence of manufacturing process such as milling, drilling and grinding, can be grouped into one product family. Some manufacturing firms group the parts by similarity of material such as plastic parts, metallic parts and integrated circuit parts. The appearance can be interpreted in dimensions (include tolerance) and features such as hole, slot.
Figure 7.2 Schema overview of customer requests

In the technical evaluation of PIMI, XMLSPY5 is used to locate any reference to support the schema dialect within the document. It loads the corresponding schema into the memory and uses its definitions to validate the XML file. If the validation succeeds, a brief message is displayed at the bottom of the main window as in Figure 7.3

Figure 7.3 A brief message of successful validation

If an error is encountered during the validation, the source of the problem is highlighted and a corresponding error message is shown as in Figure 7.4.

Figure 7.4 A brief message of unsuccessful validation
The Validation command also automatically includes a well-formedness check so there is no need to first use the “check well-formedness” command manually before validating a file.

7.4 Operational Stage

Two aspects about the operation of RPDS are illustrated in this section: operation of a system from the user’s perspective and the technical issues in the operation stage. Technical issues for implementing OBPDM, HKM and PIEM are discussed separately while the human aspect of RPDS operation is discussed generally.

7.4.1 Technical Aspect of RPDS

Two technical issues about maintaining object-oriented software need to be addressed. The first issue is that good design of the system architecture can avoid much effort on maintenance. The second issue is to keep track of the relationship between environment change and evolvable objects. There is a close relationship between object-oriented design and maintenance. Problems will arise if the design of object-oriented structure is not adequate, especially inadequate use of inheritance, dynamic bind and cooperation with small methods. Inheritance hierarchy should be not be more than three as it may impose a barrier for maintainers to find out the dependency of operation and properties within the superclasses to subclasses. Identifying the association among objects is one of the ways to find out the inherited properties and operations in the hierarchy.
Dynamic binding sometimes may cause difficulties for maintainers to determine which method should be executed. This problem can be resolved by applying methods such as the comparison of external dependency graph to detect polymorphism inconsistency. Subtle errors can also be avoided if method naming is consistent.

Deploying small methods in the system lets programmers more easily understand the individual method but it may not help maintainers to understand the whole system behavior. Tools based on analysis can provide a useful model to describe and visualize a complete chain of interaction among objects. As a coin has two sides, the advantages of using object technology in software development sometimes may bring drawbacks in maintenance. Careful design may help to prevent the problems from happening and good documentation can solve the problems.

Constructing the case base is the first step for implementing CBR and several factors need to be considered. Equilibrium should be maintained such that the case base has a sufficient number of cases to resolve the problem. On the other hand, the case base should not be too large as it may adversely affect the retrieval time. It is necessary to set the maximum capacity of the case base and record the success rate of problem solving by means of the retrieved cases. Another key issue is the case competence which attracts attention from the perspective of the so-called case base maintenance since more cases are accumulated day by day. As the business environment is changing, cases are evolving. The capability of the original cases to solve the new problems needs to be reviewed regularly. Although it is claimed that CBR can adapt the retrieved case to new problems, human intelligence is still involved to respond rapidly to unexpected changes. In the competitive environment of continuous changes, the confidence value of the retrieved case will be relatively
lower than in the common situation. Confidence value can reflect whether the retrieved cases support the solution. Confidence value of the retrieved cases can be measured in terms of similarity, typicality and deviation, that is, similarity between the retrieved cases and the query cases, the numbers of properties/compareables possessed by both query cases and the stored cases, the difference in the estimated value of each compareables selected with the estimated value of the query (Cheetham, 2002). Confidence value is found to be directly proportional to similarity and typicality but inversely proportional to deviation. Similarity plays a pivotal role in confidence level as the hierarchy of the induction tree is crucial for case retrieval.

PIEM capitalizes data exchange among business partners with the latest advances XML technology related to product development, thereby enhancing the agility of manufacturers. The proposed PIML enables efficient data exchange among various data objects which resides in different platforms by providing a meaningful semantic framework for managing the various activities during product development. Although XML has been widely deployed, the old versions of web browser and some legacy systems may not support XML. Therefore, a suitable web browser should be installed to ensure the full functionality of PDEM. Since most legacy system support ASCII files, XSLT can transform XML to ASCII file to streamline data exchange.

7.4.2 Management Aspect of RPDS

Apart from the technical aspect, the human aspect of implementing three modules needs to be considered. In RPDS, CBR, which is regarded as an aspect of artificial intelligence, is related to reasoning through analogies. Most researchers in the field of CBR may claim numerous advantages for knowledge acquisition and assimilation. However, it may impose fear for the users since they may lose their
sense of security if the computer has potential to take their place. Top management should convey a clear message that RPDS is regarded as a tool to assist them to work more efficiently. As the efficiency of the product development process is increased, knowledge workers can allocate more time to business development thereby enhancing the competitive edge of the company. As a result, the employment of RPDS does not imply that manpower can be reduced. Top management of an enterprise should not set their goal as reducing cost but as increasing revenue by means of applying an efficient information system. In order to streamline the operation of RPDS, formal training on the use of the RPDS is essential. Although the RPDS seems to be easy to use, formal training is essential to ensure the proper deployment of staff. Users of RPDS should have adequate background knowledge of CBR or they may find it difficult to understand the logic and operation flow of the associated programs. Indeed, some powerful CBR tools are not intuitive and training provided by CBR tool developers is important. As the business environment changes that cause product development activities continuously to evolve, the system will require maintenance and enhancement. Changes to requirements or additional requirements will become necessary with the passage of time. The management process of applying these changes in a controlled way is essentially the same as when the project was being developed and the documentation produced during the project phase should be maintained and kept up-to-date.

In conclusion, RPDS produces the main benefits of restructuring the relational database into an object-oriented format which produces a more effective, efficient, flexible and stable data-knowledge integration system. There are four stages for implementation during which some issues need to be taken into consideration. The start up stage is to establish a timeline with milestones and a contingency plan, to
identify the role of each participant and to prepare the network infrastructure and system environment. In the structural formulation stage, object technology is used to analyze and design the whole system. The relative attributes as well as behaviors are recognized and the associative relationships between the classes can be established.

Each object can be simply interpreted as an entity of XML <tag>. Apart from showing the static Meta data in XML, the dynamic behaviors of the object are encoded in a scripting language, and the users are allowed to customize the display style of data. The integrated knowledge-database system assists relevant staff to master codified knowledge to complete the relevant tasks. The knowledge based system relies heavily on the updated information and utilizes the algorithms to interpret knowledge. In order to streamline the product development processes, PIML is constructed by studying the users' habits and terms/vocabulary. The testing and evaluation stage is based on the hypothesis that is derived from the features of RPDS and the collection of users' feedback can facilitate continuous improvement. In the operational stage, the maintenance of the information system in the ever-changing manufacturing environment has been explored from the point of view of the managerial and technical side.
CHAPTER 8. CASE STUDY

To validate the feasibility of adopting RPDS in enhancing new product development, case studies have been conducted in three Hong Kong based companies including Group Sense Limited (GSL), Confi Metal Finishing Company and Gara Plastic Products Factory Limited. These three companies fabricate different products and adopt three different production modes including Original Equipment Manufacturer (OEM), Original Design Manufacturer (ODM) and Original Brand Manufacturer (OBM). In this chapter, company profiles and the existing practices of each manufacturer are studied and the employment of RPDS is introduced and discussed.

8.1 Case Study in Group Sense Limited

GSL is proficient in designing and manufacturing a wide range of hand-held electronic products that enable consumers to acquire and to utilize information in a convenient and fast manner for education, entertainment, data storage and communication purposes. GSL designs and manufactures electronic dictionaries, Personal Digital Assistant (PDA), translators and electronic organizers. GSL currently employs between 3000-6000 people in PRC and Hong Kong. GSL was founded in 1988 and launched the first Instant-Dict electronic dictionary EC1000 in 1989. GSL established the brand as “Instant-Dict” and started OBM early rather than transforming from OEM or ODM to OBM.
8.1.1 Existing Workflow and Challenges Faced by GSL

GSL, a major manufacturer of electronic products, invests significantly in research and development activities related to innovative product design with focus on changing customer demands. Apart from transforming the customer needs to value-added features on the new product, the company utilizes the enterprise information system to support product design, procurement, production planning and inventory management.

In GSL, the business operations require experienced product designers to cope with hundreds of product specifications every year, some of which belong to highly sophisticated electronic products. The recent product of GSL has shifted from the electronic dictionary to the PDA. The workflow of mechanical department, electrical department, and software department in fabricating PDA is shown in Figure 8.1.
Figure 8.1 Workflow of GSL
The major challenge is the design review during the early stage of product development. In the past, the data stored in proprietary and fragmented discrete database systems and platforms which were normally inflexible and required significant experience to use them. The legacy system also requires relatively high cost and long time to develop new products, affecting the progress of product development. The research and development department of GSL supports the view that XML can be used as a universal data standard where a versatile solution can be further used to produce rich and attractive product specifications with powerful collaboration capabilities.

8.1.2 Roadmap for Implementing RPDS

GSL needs to provide qualitative products or service in order to meet the increasing customer expectations where creative products can open new markets and attract the attention of potential customers. As a result, design practitioners not only need to cope with the ever-changing customer requirements but also need to co-operate with team members within the product development lifecycle for optimizing the effectiveness of the development processes. Due to the iterative nature of product design, the information or data handled by each knowledge worker is becoming more complex and profuse. A systematic data model structure can be applied to the above situation for improving knowledge sharing and dissemination. The key steps of data analysis and transformation are shown in Figure 8.2.
Object technology and PIML help to create a new product specification and evaluate the effects of design changes. The integration of the RPDS indicates the steps for objectizing the data. The implementation of each design change is done according to internal and external factors, where efficient data transmission is essential. Seamless integration with diverse back-end enterprise data systems ensures that all the information relevant to different departmental processes can be quickly accessed and assimilated in the product design process. The details of each layer within the product development lifecycle of electronic products are illustrated in Sections 8.1.2.1 to 8.1.2.5.

8.1.2.1 Analysis of the Data Structure with an Object Oriented Concept

In order to reduce the complexity of the data structure, the object-oriented modeling technique is used to simplify the data model. Inheritance and polymorphism of object technology are applied for responding to different design changes during the product development cycle. Inheritance, which is one major feature of object
technology, reduces the complexity of maintenance. Figure 8.3 is a subclass hierarchy diagram which shows the superclass (Product Information) that is the "root" of the tree. The subclasses, which are descended from the superclass, inherit the information from the superclass. The Product Brief Form is simultaneously a subclass of the Project Request Form where the Project Request Form is a subclass of Product Information. As a result, the Product Brief Form inherits all the attributes and behaviors from its ancestors (Project Request Form and Product Information). In addition, the Project Brief Form has its own attributes (comments) and behaviors (initiate).

![Diagram of object-oriented model of product development in GSL](image)

Figure 8.3 Object oriented model of product development in GSL.
8.1.2.2 Simplify the Data Structure by Inheritance

Inheritance can be performed in XML Schema. XML Schema helps formalize the data structure to express syntactic and value constraints applicable to its document instances (W3C^2, 2001). ProductInfo is defined as complex type in the first line and the attributes, which are ModelNo and Description, are the attributes of the superclass. \(<\text{extension base= "ipo:ProductInfo"}>\) is used to inherit the attributes from the superclasses (Product Information) to the subclasses (Project Acknowledge Form, Project Request, Form and Technical Analysis Report). The additional attributes owned by subclasses are listed after the tag of \(<\text{extension base}>\). To illustrate the example mentioned before, a sample code of XSD is shown in the next page.
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.3 U (http://www.xmlsp " elementFormDefault="qualified" attributeFormDefault="unqualified">-->
<xs:import namespace="http://www.altova.com/IPO" schemaLocation="C:\Documents and Settings\Administrator\Desktop\PDinheritance.xsd"/>
<xs:element name="ProductDev">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PDFlow">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ProjectAck">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="ProductInfo">
                    <xs:all>
                      <xs:element name="ProjectAccept" type="xs:boolean"/>
                      <xs:element name="EngSampleReady" type="xs:date"/>
                      <xs:element name="EstPDCharge" type="xs:decimal"/>
                    </xs:all>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="TechAnalysis">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="ProductInfo">
              <xs:all>
                <xs:element name="Department" type="xs:string"/>
                <xs:element name="EstManReq" type="xs:float"/>
              </xs:all>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
Polymorphism, which is one of the critical features of object technology, is preferred for updating the data in various objects instantly and automatically. This brings benefits that come from responding to the design change adequately, when it is applied in the product development process. Polymorphism refers to the ability to take many forms. When an object received a message that it understands, it invokes some behaviors. Those behaviors are part of the definition of the object. Various kinds of objects can understand the same message, yet respond differently (Goldberg & Rubin, 1995). Suppose the dimension of the electronic product changes from $150\text{mm}\times120\text{mm}\times15\text{mm}$ to $100\text{mm}\times100\text{mm}\times10\text{mm}$, the staff of the marketing department need to edit the catalogue and promote the product as the smallest size in the market. Before the products are sold in the market, the tool makers need to make another mould with new dimensions for the product housing. On the other hand, electronic engineers have to ensure that the circuit board is large enough to hold all integrated circuits. Various parties respond differently when the product size has been changed. Those behaviors can be encoded under different objects/classes which represent different parties involved in the product development cycle. In PIML, the feature of polymorphism is embedded in the script such that OBKIS can react to change rapidly in the dynamic environment.

8.1.2.3 Construct the Transformation Rule for Information Manipulation

XSLT, which has evolved from XSL, which is a language for transforming XML documents to another XML documents. *XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary* (W3C, 1999). To interpret the definition of XSL provided by W3C, the styling of an XML means presenting data in
a certain format. Data transformation means parsing an XML document into a tree of nodes, which is similar to DOM, and then converts the source tree into a result tree. XSLT processor can convert the XML document or well-format text to the specified format such as text, html or xml according to the template rules. The output data can be viewed by a web browser or act as a data source for a database/application.

In addition, XSLT possesses programming logic since XSLT supports numerous sets of flexible data types, a full set of operations and programming flow logic. The following example illustrates how XML combines with XSLT which captures programming logic on PIML.

The Technical Analysis Report contains the data such as model no, product description, estimated product development cost, estimated man power required, estimated material cost, estimated new equipment and tooling investment, the cost of them as well as comments. Data stored in the relational database is exported in the following XML format.
Since insufficient manpower is the critical problem of a production process, the solution and comments are stored as cases in a case base. When the manpower is more than 100 man months, the manufacturing firm may face the problem of insufficient manpower. The template rule (<xsl:when test="Estman&gt; 100"> ) is set in XSLT such that the relative data (comment) is retrieved and manipulated as the solution/advice for the case.
XSLT transforms the data in text format with delimiters and those delimiters act as anchors of the text file. The text followed by delimiters will be imported into the corresponding case of the case base where the default delimiters include Case Name:, Description:, Solution: and Questions:. The output format transformed by XSLT processor is shown below.

<table>
<thead>
<tr>
<th>Case Name: Insufficient manpower</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description: manpower cannot meet production rate</td>
</tr>
<tr>
<td>Solution: Employ more workers</td>
</tr>
</tbody>
</table>

The above text file is imported into the case base and acts as knowledge source of the CBR system since most CBR systems allow text to case conversion. The imported case can be edited and revised in the case authoring module. XSLT also can transform XML file to HTML which is more presentable for users as various parties in supply chain can access such systematic information regardless of the platforms being used.
8.1.2.4 Data Conversion by PIML to Knowledge Base

With the implemented system, users can load the database to knowledge repository and convert it to XML. Figure 8.4 shows the screens of data conversion. The screen at the top of the figure contains the data field of the technical analysis report which includes model number, product brief description, department, estimated manpower, estimated product development time, estimated material cost, estimated new equipment and tooling investment, estimated new equipment and tooling, estimated product development charge as well as comments. The screen at the top of Figure 8.4 has a list box that displays XML file in tree hierarchy. The screen entitled "PIML to Case Conversion" shows that an information-conversation procedure was invoked after establishing the link for converting XML to case. The screen at the right hand side shows that the rule can be created to filter the essential information for the case with XSLT. Having constructed the case base, users can retrieve the case from the knowledge repository and the most similar case is found using the nearest-neighbor algorithm.
1. The data in database is converted to XML

2. PIML helps convert XML to case base

3. User can insert the rule to manipulate information for knowledge discovery.

Figure 8.4 Snapshot of the screen of data conversion
8.1.2.5 Application of CBR in RPDS

Technical analysis is essential for evaluating the feasibility of new product development. The required manpower, product development time and cost are estimated by examining the product design and then applying heuristic knowledge gained through experience by considering the manufacturing processes and assembly operations. The resource estimation task is ill-structured and heavily dependent on the expert's experience. Therefore, it is suggested that CBR is deployed for estimating the engineering cost so that the accuracy of cost estimation will be enhanced; the pressure of work on one or two experts is reduced and the expertise of the expert is maintained. Table 8.1 shows the slot of the case in HKM.

<table>
<thead>
<tr>
<th>Case Name</th>
<th>Unique number generated by CBR system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case Index</td>
<td>Model No.</td>
</tr>
<tr>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Solution</td>
<td>Estimated manpower</td>
</tr>
<tr>
<td></td>
<td>Product development time</td>
</tr>
<tr>
<td></td>
<td>Material cost</td>
</tr>
<tr>
<td></td>
<td>New equipment and tooling</td>
</tr>
<tr>
<td></td>
<td>Cost of new equipment and tooling</td>
</tr>
<tr>
<td></td>
<td>Product development cost</td>
</tr>
<tr>
<td></td>
<td>Comment</td>
</tr>
</tbody>
</table>

(I) Case retrieval

Case retrieval undergoes several subtasks including feature identification, match and search and case selection. Since the problem can be input as free text and the semantics of the problem should be understood by case base reasoner, some relevant questions or synonyms related to the input problem are provided for users in
order for them to discern the system identity of the problem within its context. A noisy or irrelevant description of the problem should be avoided. Index is a computational data structure that can be held in memory and can accelerate the retrieval process. According to the criteria of quality index (such as predictive, discriminative and scalable), product type and product description are selected as case index for the calculation of engineering cost. A high degree of similarity between the source case and target case implies that there is less distance between the source case and target case. Cases are distributed within a multi-dimensional search space where each attribute implements one dimension. A structural attribute of a cube is an organized hierarchy of categories/levels which describes a similar set of members. For example, product structure is shown in a hierarchical chart that shows all the subassemblies, sub-sub assemblies, components and raw materials. The case based reasoner retrieves similar product structures and product types from the product data vault and the case based reasoner retrieves the cost of each component by matching the critical elements of the cost calculation such as direct labor, direct material, and operation cost using the nearest-neighbor algorithm.

(II) Case Adaptation

Once similar cases are retrieved in a CBR system, it will adapt the solution stored in the retrieved case to the needs of the current case. Adaptation looks for prominent differences between the retrieved cases and the current case and then applies formulae or rules that take those differences into account when suggesting a solution. Structural transformation is adopted in HKM. The most similar cases are retrieved and adapted based on three major procedures which are: additions, substitutions and deletions.

For examples, GSL designed one PDA (V888BT) which has a design similar
to that of the V888. The only difference between them is that V888BT has a small camera and use blue tooth for data transfer. The product cost is adjusted according to the addition and deletion of the subcomponent. Substitution is a more complex procedure and it needs to consider the knowledge goal to obtain information necessary for a specific adaption. OBPDM is object-oriented and hierarchically organized with the feature of adaptation which is done through polymorphism. Figure 8.5 shows the screen of a case retrieval module. Users can find out the manpower, engineering cost and product development time for designing and manufacturing similar electronic products.

Figure 8.5 Case retrieval module of CBR
Product development time needs to consider product cycle time which acts as an abstract class. Each manufacturing process has its own cycle time and this overrides the abstract class during case adaptation. E.g. The cycle time of metal cutting is the summation of cutting time, time of putting a sharp tool on the machine, time of adjusting or positioning tools on each work piece and time of loading/unloading one work piece whereas the cycle time of die casting includes the time of filling the casting, cooling and extraction of the component from the die. As a result, CBR estimates the product development time and engineering cost by identifying the manufacturing process of each component. The adaptor of the CBR adapts the cases with polymorphism with reference to the information migrated from a database in the form of PIML.

8.2 Case Study in Gara Plastic Products Factory Limited

Gara Plastic Products Factory Limited (a toy manufacturer) is an OEM which also has the vision to become an ODM. Gara also provides designs for clients in addition to its manufacturing function. Gara has worked with Disney for more than 19 years and has manufactured over 500 Disney characters. In addition, Gara has produced Warner Brothers characters (over 100 collections) and Sanrio characters, as well as Power Rangers, VR Troopers, Spiderman, X-men, Dragon Ball Z, Smurfs, Garfield, Snoopy, and many other internationally well-known cartoon characters.

To reduce operation costs and stay competitive, Gara has set up production facilities offshore, mainly in the PRD and the top management wants to shift parts of the job of product development to the PRC staff in order to reduce the workload of HK staff. However, data decentralization causes work duplication which becomes a barrier to shortening product development time. An efficient data exchange standard
between HK and the PRC is required and a systematic method to synthesize the product development model is essential.

8.2.1 Existing Workflow and Challenges Faced by Gara

Due to the keen competition, short product life cycle and changing market trends, manufacturers are dedicated to their own core value-added processes. Companies attempt a balance between OEM, referred to suppliers who produce according to specified customer requirements and own brand production. Gara, which is one of the OEM, face the challenge stated before. A paradigm shift among OEMs has occurred. They have changed from manufacturing with only the buyer’s product specification to go on developing a complete product design from merely a design concept prior to production. This is expected to continue into the next century and they are known as ODM. Gara provides customized designs for various clients to fulfill customer requirements and to keep up with the market trend. Similar to most OEMs, Gara usually redesigns the product by adding new product feature to the old product. However, their existing capacity is a constraint because the existing equipment may not produce the new product features needed for the new product. Another challenge faced by the manufacturer is the shortness of the product development cycle. Delivery time, which is around 45-90 days on receipt of a letter of credit, depends on the complexity of the product and the availability of material. The short delivery time causes the engineers to use their past experience and knowledge rather than plan and analysis in detail to accomplish each order. Design agility and manufacturing flexibility are the critical success factors which help to resolve these problems. It is found that product development is the value-added process. The existing workflow is shown as following.
Product development entails:

i. Design

ii. Model making

iii. Tooling

Figure 8.6 shows the product development workflow of Gara between various parties.

Figure 8.6 Sequence diagram of product development in Gara

8.2.2 A Roadmap for Implementing RPDS

The development of a new RPDS is not just for validating the feasibility of AI techniques in design activities, it also helps enhance knowledge management. Those companies which can leverage knowledge to introduce new product earlier than their competitors can gain a greater share in the market. The case study below is
used to show how RPDS operates in an ODM such as Gara. The implementation is undertaken module by module.

8.2.2.1 Construction of Object Model in OOPDM

In the object model, ‘initiate’ and ‘approve’ are the behaviors of the object (General Information). When the document is initiated, new forms, which are the Design Drawing Start Form (DDSF), the Prototype Start Form (PSF) or the Tooling Start Form (TSF), are created. When those forms are approved, the status of the documents is changed. Behavior, which is identified as dynamic, may bring status change for the objects. The associations between objects are represented as vocabularies to indicate the interaction of the objects shown in Figure 8.7.
Figure 8.7 The object model of Gara
8.2.2.2 Transformation of Knowledge from Database in HKM

The design processes of a new product, which usually involve the marketing, R&D and engineering departments, require interaction between various departments. However, few organizations recognize the importance of interaction and shift the responsibility of collaboration to the project coordinator. The project coordinator usually becomes the bottleneck of information if the workload of data is more than that he can handle. The decision to design a new model is made if the marketing department requests that a new product is introduced or designers could transform customer requirements into innovative ideas. The designers may reuse the past designs or modify them based on their experience in considering whether the parts can satisfy the new requirements of customers and achieve a certain quality level. Indeed, the design practitioners may memorize the past cases which are similar to the current one according to various factors including the customers’ descriptions of what they need, the features of the past products related to customer’s satisfaction, telemetric data collected in past editions of the manufacturing process, the quality of past similar products and the result of R&D projects to refine the product features.

(I) Use of CBR to leverage design knowledge

Designers can capture information that is critical to the creative process in product development. After designers have entered the description of the problem, a list of cases along with questions based on the users’ entries is displayed as in Figure 8.8. By retrieving the matched case, the system helps the designers to master updated information and provide some insight for them to design the toys.
Figure 8.8 Problem resolution of RPDS

Since the past designs are usually feasible to manufacture, there are no obvious design problems. Besides, it is recognized that developing a new version of the product design from scratch would take far too long. Therefore, the past CAD drawing with a useful description can help designers develop innovative designs for the target customers in a short period of time. Figure 8.9 shows that users may simply input the product features required by the market trend such as stylishness and good surface geometry. The CBR system retrieves the related past designs with CAD drawing and proposes a new solution for the designers.
CASE NAME: Stylish Sedan  
DESCRIPTION:  
Customer prefer a stylish and good geometry car.

SOLUTION  
The conceptual design is developed in R&D  
Product ID: S_028  
Product Family: Sedan_01  
Surface roughness less than 5 micro inches  
R&D has started to conduct a research named "Be Accurate" on 1/5/2003 and they  
develop low stress grinding to have a good geometry and excellent repeatability of  
surface finish. It is expected to be completed in 10/6/2003.

Figure 8.9 CAD library within the case base reasoning system

(II) Integrating CBR and relational database

The solutions retrieved by the CBR system are usually composed of  
different data in the relational tables and any activities and decisions can change the  
associated data which in turn may alter the case solution. Figure 8.10 illustrates how  
to make use of the corporate data in the relational database to help knowledge workers  
master the latest information for the purpose of reducing the redesign work. Database
in HKM may contain some useful and up-to-date data such as

i. The product specification

ii. The product family to which it belongs

iii. Product ID

iv. The activities related to product development

![Diagram](image)

Figure 8.10 Co-relation between the case base and the relational database

Some critical information such as the activities of other departments may lead to a change of product size. In the case example, low stress grinding, which comes from the “ProcessDesc” field of the “manufacturing” table, allows the size and dimensions to be held to a much closer tolerance than in conventional machining. The
car body can be produced with very accurate surface geometry after the project "Be Accurate" has been activated. "Be Accurate" is one of the projects from the table called "Project" within the corporate database. When the project completion date is earlier than the current date, new data of the product size is updated. And the roundness can be maintained within 0.00001", which comes from R&D's report. DDIS synchronizes the data of case base with the relational database and reduces the chance of their being an information gap. Product data is exported from the relational database to the case base and the data mapping between corporate databases to the knowledge repositories is shown in Table 8.2.

Table 8.2 Data mapping between corporate database to knowledge repositories

<table>
<thead>
<tr>
<th>Corporate Database</th>
<th>Knowledge Repositories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name</td>
<td>Field name</td>
</tr>
<tr>
<td>Manufacturing Process</td>
<td>ProcessDesc</td>
</tr>
<tr>
<td>PartDetail</td>
<td>PartID</td>
</tr>
<tr>
<td>Project</td>
<td>StartDate</td>
</tr>
<tr>
<td>Product Family</td>
<td>CompDate</td>
</tr>
<tr>
<td></td>
<td>FamilyID</td>
</tr>
</tbody>
</table>

8.2.2.3 Deployment of PIML in PDEM

XML is a markup language proposed by the World Wide Web Consortium. Similar to HTML, it can go straight to be used on the internet. XML is comprehensively used due to its compatibility with other SGML such as HTTP. PIML supports a wide variety of applications and it allows design practitioners to perform (I) Authoring, (II) Browsing and (III) Content analysis.
I. Authoring

To author XML, Notepad or some commercial XML software tools such as XMLSpy or XMetal can be used. They can provide an integrated development environment for the XML that includes editing and validation of XML Schema/DTD and XSLT. Having constructed the object model (Figure 8.7), system analysts can start authoring XML. Project engineers for example can carry out different tasks with various classes. Project engineers initiate the project and fill in a new design drawing start form. As designers complete the drawing, the project engineers will contact the prototype makers and tool makers. In the product development cycle, project engineers play various roles and interact with different parties. The following XML Schema illustrates the role performed by project engineer and the association between project engineers and related parties.

```
<xs:complexType name="projectengineer">
  <xs:all>
    <xs:element name="dept" type="xs:string"
      minOccurs="1" maxOccurs="1"/>
    <xs:element name="CEng" type="xs:boolean"
      minOccurs="0" maxOccurs="1"/>
    <xs:element name="initiate">
      <xs:complexType>
        <xs:element>
          <xs:complexType name="contact">
            <xs:complexContent>
              <xs:sequence>
                <xs:element ref="PrototypeMaker"/>
                <xs:element ref="ToolMaker"/>
              </xs:sequence>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
      </xs:complexType>
    </xs:element>
  </xs:all>
</xs:complexType>
```

In the XSD code, the object called 'project engineer' is defined as a complex type. The data type, minimum occurrence and maximum occurrence are defined. In the real situation, project engineers need to contact the prototype maker prior to
contacting the tool maker. The contact sequence can be quoted in the <xsd:sequence> tag.

There are two special associations of object modeling known as inheritance and polymorphisms which are the characteristics of object-oriented modeling. Inheritance is the feature of object technology to reduce the complexity in the maintenance process. The object model of product development (Figure 8.7) shows that general information is a superclass where DDSF, PSF, TSF are the subclasses. DDSF, PSF and TSF inherit all the attributes and behaviors from the General Information. The lower part of the screen in Figure 8.11 shows the attributes of DDSF. The highlighted attributes are the attributes inherited from GInfo.

Figure 8.11 Constructing XSD with XML Spy
To represent the data structure in XML, XML Schema is deployed which is an XML language for describing and constraining the content of XML documents. A XML Schema specifies the elements that can be used in XML documents along with the structure that these elements must follow. In the case example, Figure 8.12 shows that the Inheritance_F.xml is valid for Inheritance_F.xsd.

![XML Schema Diagram](image)

**Figure 8.12 Validate the XML file with XML schema**

Pursuing the information infrastructure that can be constructed by object oriented modeling, the XSD has defined the document specification; accordingly, the object oriented program can structure dynamic behavior for PIML. In the design process, Gara needs to identify what kind of drawing is required by customers. There are two typical drawings, a technical drawing and a presentation drawing in the
workflow of product development. Project engineers usually can acknowledge the exact delivery date for those clients who require a presentation drawing. The number of days required for completing the presentation drawing depends on the number of days needed to collect customer requirements and the number of days the designer needs to finish the draft.

However, it is more complex to determine the number of days required for completing technical drawings since it is necessary to state tolerances in the technical drawings. Tolerance may sometimes depend on the equipment/machine capability. As a result, it is usual that project engineers cannot state or tell the customers the exact completion date of technical drawings. In the process of developing a technical drawing, collecting customer requirements and preparing the conceptual drawing have to be done beforehand. The draftsmen produce technical drawings based on the conceptual drawings from the designers. Project engineers get the notification from draftsmen and tell the customers how many days it will take to complete the draft. If the time for developing the draft is less than a month, project engineers may notify the customers that the technical drawing can be delivered within a month. Otherwise, project engineers tell the customers that they will need to wait more than a month for the drawing.

The above situation can be represented by polymorphism which is the characteristic of object technology. The class, presentation drawing, has attributes "The number of days required for collecting customer requirement" and "The number of days for finishing the draft". The presentation drawing class also contains the function (suggest) which shows the days required for preparing a presentation drawing for the customer.

Since the operation of preparing a technical drawing is quite similar to that
of preparing a presentation drawing, it is also necessary to consider the number of
days required for collecting customer requirements and the days for finishing the draft
and for giving suggestions to the customer. Therefore, the technical drawing class
inherits the properties from the presentation drawing class. However, as mentioned
before, the process of preparing a technical drawing involves various parties and it is
not easy to tell the customers the exact delivery date. Therefore, the suggestion
function of the technical drawing class will override the presentation drawing class.
Instead of giving the exact delivery day, it only shows the message "Customer need to
wait more than 1 month" if delivery day is more than 1 month or else "Draftsman will
finish it within a month" if the delivery day is less than 1 month. The user interface is
shown in Figure 8.13

Figure 8.13 The interface of the design drawing start form
II. Browsing

Those data or document in XML format can be used across any platform. For example, if a design practitioner uses Mac PC to design product and would like to access product information stored in Window platform, he can access the product data in XML format with a browser through the intranet.

Since XML is not good for presentation, XSLT has evolved from the early XSL standard. XSLT specifies a language definition for XML data presentation and data transformations. Data presentation means displaying data in some format and/or medium while data transformation means parsing an input XML document into a tree of nodes, and then converting the source tree into a result tree. Transformation is about data exchange. After transformation, the page is represented in HTML as in Figure 8.14.

![XML to HTML Transformation](image)

<table>
<thead>
<tr>
<th>ProductID</th>
<th>Product Name</th>
<th>Product Category</th>
<th>Subcategory</th>
<th>Product Description</th>
<th>Light</th>
<th>Battery</th>
<th>Sound</th>
<th>Price</th>
<th>Picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>Bedroom Play set</td>
<td>Play set</td>
<td></td>
<td>Ft all 11 1/2 (Max) Fashion Doll Fireplace, Per pet set, Books, chair, Round Table, Table Lamp, Clock, Candle Holder and Frame</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Bedroom Play set</td>
<td>Play Set</td>
<td></td>
<td>Dinning table, chair, Bed</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Dining Room Play Set</td>
<td>Play Set</td>
<td></td>
<td>Dinning table, chair, Dining, Table, Glass</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Wardrobe Play set</td>
<td>Play Set</td>
<td></td>
<td>Full-Length Mirror, Bi-fold mirror, and 3 hangers</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Ladder Playset</td>
<td>Play Set</td>
<td></td>
<td>Info, chair and end table</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>New Bedroom Play set</td>
<td>Play set</td>
<td></td>
<td>Bed, light with battery, table</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Maple Mirror</td>
<td>Accessory</td>
<td></td>
<td>Portable Doll and Cradles Mirror 1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Coat Doll 1</td>
<td>Figure</td>
<td></td>
<td>Nest Head, Footed Hair, Tying Waist, Movable Arms &amp; Leg Dolls in Orange Dresses A, sewing set and Dolls</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Coat Doll 2</td>
<td>Figure</td>
<td></td>
<td>Nest Head, Footed Hair, Tying Waist, Movable Arms &amp; Leg Dolls in Pink Dresses A, sewing set and Dolls</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8.14 HTML is formed by serializing the XML file with XSLT
III. Content Analysis

Design practitioners can access a tremendous amount of information from various sources. The main concern is how to extract critical and meaningful information to make decisions. Since most Hong Kong OEMs are SMEs which have limited resources, the spreadsheet is the most common tool they use for content analysis. In this example, the data retrieved from the database is exported as XML format via a web server. There are some commercial spreadsheets (such as Excel 2002) which can open XML via the web server in order to achieve data analysis. As Gara sets up the web server in HK and XML data is published in the website via Web server the staff of Gara in PRC can type ‘http://gara/XML/PI.xml’ in the web address. Having connected to the web server and retrieved the data from the database, the data is displayed in the spreadsheet as shown in Figure 8.14.

The design practitioners can click the data menu and select “Pivot table and pivot chart report” to conduct a simple analysis. For example, a project engineer would find out what kind of product category has been bought from the market at the highest cost. Design practitioners can drag a category to the left column and then drag the product name next to it. The price field is dragged to the cell above category (A3). The project engineer can find out the price of each product and the total price for each category in the ‘total’ column. To enhance the progressive introduction of product design knowledge associated with decision support functionality, a database interconnecting with the knowledge base system is proposed for Gara. RPDS is deployed for accessing data from a database via the web server and formatted as XML data. XSLT can serialize the XML as text format which is one of the data source formats accepted by a case based reasoning system. Table 8.3 shows the process of accessing data from an SQL database to a commercial case based reasoning system.
Table 8.3 Process of migrating data from database to case base

1. Product data is stored in SQL database

2. SQL script is embedded within the XML which form a template file to retrieve product data from SQL to XML via web server.

3. Browser displays the product data in XML format.

4. XML is exported as text file with XML Spy

5. Text data source is imported to case based reasoning system through ODBC.

6. Design practitioner can retrieve the product with the highest potential for the market through case based reasoning system.
8.3 Case Study in Confi Metal Finishing Company Limited

Confi Metal Finishing Company Limited (Confi) was established in November 1990 and specializes in precious metal plating including electrolytic gold, nickel, palladium-nickel and rhodium. Its administrative office is situated in Hong Kong with two manufacturing plants located in Chopo and Ho-Au of Shenzhen, PRC. respectively. A total of over 2000 employees are being employed in two factories in Shenzhen, comprising high-caliber staff, managers/supervisor in different divisions, account executives, skilled and factory workers. Confi’s factory in Shenzhen can accommodate and produce a minimum quantity of more than 1,000 racks per day operating at a maximum of 24 hours a day, 6 days a week in Chopo Factory or Ho-Au.

8.3.1 Existing Workflow and Challenges Faced by Confi

Apart from accepting job orders for electroplating, Confi also fabricates electroplating equipment/tools. Recently, Confi is concentrating on the vacuum ion plating process. The workflow of Confi shown in Figure 8.15 is masking, cleaning and pre-treatment, plating, quality control and finally painting/spraying and storing in the finished goods warehouse.
Figure 8.15 Production Workflow at Confi
Quality control of the plating process receives great attention from Confì's top management. Data analysis is critical and essential for continuous improvement of the process; and defining electroplating-related terms provides the impetus and glue towards a shared culture approach to leveraging knowledge. Similar to most Hong Kong based manufacturers who set up factories in PRC, data exchange among headquarters and production plants is difficult. Manufacturers need to face the challenge of responding to turbulent market changes, meet escalating customer requirements and produce good quality products within a short product development time. The shorter the time-to-market, the greater market share the company can gain. PDM is therefore essential as it helps to let the people know enough about the product development environment such that they can deal with changes with updated information efficiently and effectively.

Acting as a technical based manufacturer, Confì regards the quality control of the plating process as one critical aspect that needs further research and development. As a result, knowledge related to solutions for technical problems and to successful parameter setting should be captured and stored in order to provide efficient solutions for future use.

### 8.3.2 A Roadmap for Implementing RPDS

RPDS has been demonstrated to show how it helps Confì to enhance product development activities. The following procedures illustrate how the present system works.

Step 1: Define XML Vocabulary. PIML is a set of XML vocabulary which acts as XML schema for product data model that allows data exchange between customer and enterprise. Define the major elements, attributes, data types and constraints on a
complex content definition.

Step 2: Build up the cases to make a case base. During the product and process design, experienced design practitioners usually recall the past cases and modify the previous design cases. Updated product data about customer requirements and design changes as well as some explicit knowledge can solve new problems.

Step 3: Replicate the data from a relational database to the case base. A mapping mechanism between the relational database and the knowledge repository has been mentioned in Chapter 5.1.3. The ultimate aim of such an approach is to achieve the automatic updating of information on cases when there is any change of data in the relational database.

Step 4: Case based reasoning is proposed as an intelligent operation for capturing knowledge in engineering design. The engineers input the problem encountered in the electroplating process. The case retrieval group box (upper left of Figure 8.16) is the list area of messages from the search agent which finds the relevant cases in the case base. The text box (lower left of Figure 8.16) is used to display the suggestion. The information comes from the solution of the case which is replicated from the relational database. The text box in the lower right hand part collects the user's recommendation on the retrieved case so that it can be used to evaluate the competence of the retrieved case.
According to the needs of Confi, it has been proposed that CBR be implemented for price quotation. The estimated quoted price will be based on the material cost of PNP/ gold, scrap material, handling fee and profit margin. The cost of PNP is based on the weight of the substrate, the thickness of metal deposited, K value of precious metal and the market price of the precious metal. Apart from considering the factors used in PNP plating, the utilization rate of the gold is also counted for those products plated with gold since the gold can be reused in the plating process. The cost of scrap material is the product of material cost, recycle rate of gold and the non-recycle rate of material.

After the design and development of RPDS which is implemented in three Hong Kong based manufacturers, further description and discussion on the RPDS would be provided in the next chapter.
CHAPTER 9. DISCUSSION

Decisions made during the early stage of product development process are based on personal experience and knowledge. Knowledge is accumulated and gained by the staff rather than the company. The turnover of experts may lead to loss of company expertise. A knowledge based system is used by enterprises in the product development life cycle to enhance their efficiency in solving engineering problems. The early part of the product development process is characterised by iterative behavior and there is little concern about information system modeling for handling design change or product customization. Traditionally, various functional disciplines have their own database storage systems, using dissimilar data formats and even heterogeneous platforms. This creates problems in data exchange among enterprises due to the incompatibility of the data formats involved (Chung & Lau, 2000). The objective of this research is to develop a "Responsive" system named the Responsive Product Development System (RPDS) with a combination of existing technologies to achieve the capturing and optimum deployment of knowledge, thereby improving the speed and effectiveness of the product development process. RPDS handles product changes and engineering changes such that it can enhance the full capability of proprietary product designing tools with the support of universal data exchange.

9.1 General Discussion

Test results of the RPDS system prototype indicate that the three modules including HKM, OBPDM and PIEM can be linked together in terms of information exchange to achieve agility, self-learning and customizability. RPDS helps to shorten the product development time by reusing the past successful cases stored in a
knowledge repository and provides a reconfigurable infrastructure by means of object technology. Capturing the essential data and converting them into useful information to support product development is the prime responsibility of product manufacturers. However, the customers’ expectations are constantly changing due to the volatile market environment. In general, maintaining the core competence is an important issue to be addressed and its achievement requires a good collaborative environment for the product development processes. A pilot application for resolving the conflicts in data exchange with the proposed PIEM has been conducted and a specific set of XML for product development initiated by the authors. PIML takes advantage of the well-structured PDS to provide a well-understood syntax and self-defined markup language to suit the particular needs of product data exchange. In particular, PIML provides a universal product information exchange standard to enable the knowledge workers to acquire sufficient and accurate information.

9.2 Discussion of OBPDM

OBPDM makes use of the distinctive features of object technology including inheritance, encapsulation and polymorphism which are favourable to the reuse and maintenance of the information system. At the present stage, the prototype framework has been developed and progress so far is encouraging. In the past decade, a paradigm shift in manufacturing has led to one that is based on speed, fast-responsiveness and agility. Object technology is used to organize, manipulate and process the information according to real-world objects that the information describes. Objects are basically “self-contained” units that encapsulate the information, and algorithms to perform their work. The introduction of inheritance can reduce the complexity of the product structure and define common operations in a superclass. As changes may be triggered
by new customer requirements, modification of the process is required. Design change is usually regarded as troublesome and as a fire-fighting issue. However, design change indeed can provide opportunities for design improvement and bring up innovative ideas for the products. Timely Reconfigurable Method Invocation (TRMI), which can be instantiated and perform individual operations under a specific subclass, provides agility for enterprises to deal with unanticipated issues. From the illustrative example of TRMI in Chapter 4, it appears that TRMI is suitable for catering for design change in a dynamic market. Some basic concepts of temporal logic are included in a sequence diagram of product development processes as time also needs to be considered. The duration of each product development activity is set for fulfilling the tight production schedule.

9.3 Discussion of HKM

The marketplace is constantly-changing. The major challenge of using such a system is to capture a vast number of tedious processes and a vast amount of project knowledge in the form of data objects since the lifecycles of existing products are decreasing and knowledge captured now may be obsolete in the not too distant future. Continuous renewal of the vast amount of knowledge is a crucial task and adequately trained knowledge workers are needed for such a process.

In order to continuously renew the knowledge in the knowledge base, the knowledge based system needs to archive the data in such a way that the knowledge representation can be supported by the updated data. Learning through adaptation is reflected by the ability to modify the solutions of the past cases to resolve new problems by introducing logic or heuristic rules. It is difficult to create a complete
decision tree that can cater for future practice so that an adequate case can be retrieved for each new query.

In spite of this positive evaluation, some constraints of HKM were found during the initial stage. There was an indication that manual data mapping is time-consuming and difficulties arose due to the automation of the data mapping process. A team of experts, who are familiar with the knowledge representation of the case base and the data structure of the relational database, is needed to design how the data-field is mapped from the relational database to the case base. Since one case consists of numerous values of data from fields in different tables, cases of CBR are cross-connected with data records in the corporate database. The interconnections are complicated and more resource need to be allocated for constructing the links between heterogeneous repositories.

9.4 Discussion of PIEM

Product design is regarded as an iterative process as many unknown factors permeate the design process. Early product definition provides a concrete description of the product which reduces misunderstanding among various disciplines and minimizes the need for redesigning or reworking. XML allows information integrity and the people involved can make consistent decisions. PIEM addresses a unique standard based on PDS for product development and presents our efforts to develop PIML, allowing data exchange between the traditional relational database and the knowledge base. PIEM allows various knowledge management tools to access, manipulate and exchange product data formatted with the universally-accepted XML specifications regardless of operating systems or platforms. Rapid product
development is implemented on top of the proposed PIML which helps data exchange
in the distributed industrial environment. After the specific information for product
development has been identified, PIML is able to customize the product data for
knowledge workers and allows them to obtain sufficient data and accurate data, no
more and no less, that reflects the real phenomenon in the market.

9.5 Discussion of the Case Study

Three case studies are implemented to validate the feasibility of implementing RPDS in OEM, ODM and OBM respectively. It is often the case that knowledge workers are suspicious of AI tools which imitate human intelligence to resolve the problems faced in daily work. One of the interviewee who is mechanical engineer of GSL thinks that human beings are valuable as they are flexible and have the ability to consolidate different sources of information to solve new problems and this cannot be done by computer or AI tools. The sales executives of Confi also doubted the accuracy of the quotation price for products and it was also revealed that they fear AI tools will replace them if the tools can do their job duty. As a result, they were reluctant to make use of the AI tools initially. As users have realized that the purpose of implementing AI tools is not to replace knowledge workers but to assist them to work more efficiency and allow knowledge workers create more knowledge or have more time to formulate business strategies, they started to use the tools and realized the effectiveness of the proposed system.

Qualitative measurement instead of quantitative measurement is adopted since there is difficulty to implement the whole system in three companies within a short period of time. However, the feedback and comment from top management towards the RPDS are positive and promising. Equipped with a robust and cost effective suite
of porting capabilities, the CBR system under RPDS also allows the design team to work in the windows environment with which it is most familiar and HKM provide up-to-date solutions for the designers to design innovative products. Referring to the feedback of the users of three reference sites, they found that HKM helps design practitioners in searching for the relevant past records. The test companies believe that OBPDM, which can model complex association among different parties and the related processes in product development lifecycle, let them track the changes more effectively in the concerned processes. PIML in PIEM allows design practitioners to define the syntactic structure and partial semantics of XML document efficiently. Since XML is compatible with numerous heterogeneous systems, test companies realize that PIEM can access the data easier and in a timely manner with PIML such that they can have a better communication with their business partners. As a result, RPDS greatly reduces the failures and barriers that hinder the progress of the whole product development. However, the solving engineering problems and reducing response time to customer requirements is only slightly improved after adopting of RPDS. The test companies found the RPDS can help to minimize the design problem but it cannot avoid the occurrence of engineering problems. Engineering problems need to be resolved by analyzing the manufacturing system, plant layout, product structure and production capacity. Agile manufacturers need to collaborate with other partners and tasks cannot be done alone. The performance of RPDS in responding to customer requirements is satisfactory. Although product data schema of PIEM provides guidance for design practitioners, precise customer specifications need to be collected and manipulated by experienced service engineers; the trends of the market and the discovery of potential markets depend on the vision of business development managers.
In summary, three manufacturing firms realize that RPDS can provide continuous improvement in overall performance. Most of the improvement is moderate and RPDS does not have a great impact on the test companies.
10.1 Summary of Research Work

The product design activities in the existing practice rely heavily on expert experience and insights which can be subjective and biased for one reason or another. The aim of this research is to propose a systematic product information system to be used at different design stages in distributed manufacturing environments. The principle and structure of a responsive product development system featuring the incorporation of OBPDM, HKM and PDEM have been developed to shorten product development time and capture intellectual capital. The unique feature of this RPDS is concerned with the approach, which focuses on the presentation of a customized and integrated view of a company’s knowledge sources to support collaborative creation and management of product development knowledge across the dispersed manufacturing environment from product concept to product disposal. This greatly enhances decision-making by providing immediate access to key business information.

(I) The object technology approach is viable to achieve agility and reconfigurability

A product information exchange tool has been built in an attempt to match the business activities with the daily-operated data in an extensively changing enterprise environment using object a model for the representation of the processes and the relationships between data objects in product design environment. Emerging technologies including object technology and case based reasoning are employed to support the infrastructure of this system.
(II) The interconnection of the relational database to the knowledge repository helps continual renewal of the knowledge of the knowledge based system.

This system is favorable to the progressive introduction of adaptive learning reasoning to the entire design operation, thereby enabling the possibility of building a learning organization, which captures experience and insights through processes regardless of the influence of manpower turnover factors.

(III) The universal data exchange standard-PIML is essential for achieving efficient product data management.

The proposed generic model, which supports rapid product development, capitalizes on the latest advances in XML related to product development, thereby enhancing the responsiveness of the changing customer requirements. The proposed Product Information Markup Language enables efficient data exchange among various data objects which reside in different platforms, by providing a meaningful semantic framework for managing the various activities during product development.

In general, the distinct features of RPDS include (1) agility for handling unanticipated change that occurs in the early stage of the product development cycle, (2) self-learning for transforming data between heterogeneous systems, (3) customizability for fundamentally construct the data schema to handle meta-data for workflow innovation.

10.2 Contribution of the Research

The reported methodology is a step toward the development of a generic model for effective communication with customers or internal information exchange in order to achieve agility, self-learning and customizability, and therefore attain the
goal of responsive product development. The major contributions of this research are shown below.

i. It is a significant contribution to formulate a new product data schema for the initial phase of product development. This new data schema provides a standardized data exchange format among manufacturing firms and their business partners. PIEM provides a mechanism and paradigm in which data exchange is no longer bound to certain platform and operation system. PIEM helps design practitioners to leverage context-sensitive information in accordance with their self-defined vocabulary and definitions.

ii. The unique feature of TRMI is the support to invoke various behaviors for the same message and override the pre-defined inherited operation such that a flexible correlation can be formulated in the iterative product design process. TRMI is characteristic of the reuse of standard procedures for product development activities, extended from the concept of polymorphism, to adapt to the dynamic market by identifying the fundamental method application issues and assessing the current state of the art in product life cycle management.

iii. In particular, HKM provides a universal product information exchange mechanism which enables the knowledge workers to acquire sufficient and accurate information. The suggested approach achieves a seamless data interchange based on neutral data files which serve to transform the corporate database into a case repository.

iv. This research introduces temporal logic to study the dynamic aspect of the object model such that the time dimension is added to the object model. Temporal logic, a kind of modal logic concerning the logic of futurity, is used
to define the possible outcome during the product design process in this
research. By having better control or prediction of the possible variations in
the product development lifecycle, the manufacturers can achieve faster
innovation, time-to-market and time-to volume.

v. This research introduces a case objectized method to support distributed
knowledge capture, refinement, reuse and sharing. The proposed
methodology aims to empower product design practitioners to directly elicit,
codify and organize knowledge. Its focus is to present more detail of CBR,
by identifying the variation between retrieved cases and query cases as well
as performing machine learning for problem resolution. To improve the case
retrieval performance, inductive retrieval is supplemented with a feature
selection method for classifying the cases into different clusters. The primary
approach is to deploy deontic logic for case adaptation by imposing
constraints for forward chaining reasoning.

10.3 Future Research

Based on the experience in the RPDS research, future research could focus
on the following areas:

i. As the advanced technology continuous evolves, agile manufacturing
enterprises may change their decisions because they receive new information
or find a new data source. Since the knowledge representation schema is
evolving, it may be different from the original one defined before. RPDS is
concerned with how to update the data from a database to knowledge
repositories; but schematic change in the knowledge repository after
completion of the system design, has not yet been considered. RRDS

10.4
therefore is limited to the designation of a flat data schema of the case base.

Further research should be conducted on evolvable objects as this may also
help us to understand information system adaptation within the turbulent
business and manufacturing environment.

ii. It is usually assumed the greater the similarity between the queried case and
retrieved case, the greater the confidence in the retrieved case. However,
Cheetham (2000) mentioned that similarity only measures the similarity
score of the surface features of the case. The retrieved cases may deviate
from the practical solution in a real situation and such deviation can greatly
reduce the confidence in the retrieved case. The case based reasoning system
in RPDS is limited in its ability to indicate confidence in the case as it is
based on common assumptions. The indicators (adaptation accuracy,
adaptation consistency, uniqueness) are suggested for measuring this
confidence. Yet the fitness of each retrieved case still needs to be judged by
the users.

iii. Further research on the reliability of data exchange between the knowledge
base and the database is required in order to enhance the effectiveness of the
whole system.

iv. It is important to examine how well the structure of the object model in
OBPDM maps into the actual software in the product design environment.
The OBPDM is based on the notion that design practitioners are event driven
and can be modeled using sequence-diagrams or state-transition diagrams.
Although those object models are supported by the literature survey, the
concept still needs further validating by studying the robustness of the system
step by step.
v. Since most manufacturing firms may implement Enterprise Resource Planning (ERP) system, Customer Relationship Management (CRM) system or Supply Chain Management (SCM) system, the integration of the existing legacy system into RPDS could be further investigated.

On the whole, this proposed system paves the way for an alternative approach to deal with the issues related to the seamless information exchange between a corporate database (normally relational) and a knowledge repository (normally system-specific format), both of which do not communicate with each other directly. The proposed system serves to enhance the progressive introduction of product design knowledge associated with decision support functionality and therefore provides a platform for further research in terms of knowledge acquisition and management.
Appendix A

Generic specification for product design—Product

Information Markup Language

The scheme includes a project and mission statement, customer expectation list, product target design specification and has references to documents describing functional and product concept characteristics. Figure A.1 shows that PIML transfers data to support the following product development processes:

i. Needs recognition—Information analysis

ii. Design specifications—Benchmarking

iii. Conceptual Design—Concept selection and concept generation

iv. Detail Design—Prototyping and Cost estimation

Figure A.1 Generic workflow of product design
(I) Needs Recognition

It is important to realize the customers' needs at the early stage of product development. To find out the implicit and explicit needs of customers, manufacturers need to carry out the following tasks which are (a) stating the mission of the project and (b) collecting information about customer expectations.

(a) Stating the mission of the project

Each project should start with a mission statement and define a clear direction for team members. Project and Mission Statements help the project team to define the goal, direction and degree of urgency that accompany the development of a new product. The major components in the Project and Mission Statements are shown in Figure A.2.

Figure A.2 Schema overview of Project Mission Statement
In the Project Mission Statement, design team representatives clarify the objectives of the product development and describe the product feature and the potential users including the primary market and secondary markets. The major elements of the project statement are listed in Table A.1.

Table A.1: Elements of project mission statement

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Title</td>
<td>The title of the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Team name</td>
<td>The design team that is responsible for the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Product Client</td>
<td>List name of client company name, address</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Product description</td>
<td>Product Name with new or special feature</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Key Goal</td>
<td>The primary goal of the product/process development</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Primary Market</td>
<td>Potential primary users</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Secondary Market</td>
<td>Other potential users</td>
<td>String</td>
<td>N</td>
</tr>
<tr>
<td>Assumption</td>
<td>Special features of the product</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>Someone who has an interest in or invests in the project</td>
<td>String</td>
<td>N</td>
</tr>
</tbody>
</table>
(b) Collecting information about customer expectations

It is important to recognize the customer expectations clearly and thoroughly before launching the product design activities. A needs recognition should be carried out by acquiring information by conducting an extensive examination of the market data, consumer reports, in-depth interviews and customer questionnaires. Figure A.3 includes the data fields that collect the “voice of customers”.

![Schema overview of customer expectations](image)

Figure A.3 Schema overview of customer expectations

Customer expectations can be listed out and priorities should be assigned such that team members can dedicate themselves to studying those critical requirements expected by consumers. As a result, a customer expectation list is designed stating the customer expectations and 5-scale ranking is defined to clarify which features are the most desirable (See Table A.2).
Table A.2 Elements of customer expectations

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Date of creating this document</td>
<td>Date</td>
<td>Y</td>
</tr>
<tr>
<td>Team name</td>
<td>The design team that is responsible for the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Client</td>
<td>Name of client company</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Prepared By</td>
<td>Team member who is responsible for this activities</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Expectation List</td>
<td>A list of expectation</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Affinity Group</td>
<td>A group of interviewees who come up with a consensus view on a subject</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Expectation</td>
<td>The expected, spoken, unspoken, and unexpected requirement</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Ranking</td>
<td>5-scale ranking is given as below</td>
<td>Integer</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>1- Undesirable feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2- Unimportant feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3- Feature would be nice to have, but is not necessary</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4- Highly desirable feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5- Critical feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comment</td>
<td>Comments that relate to customer expectations</td>
<td>String</td>
<td>N</td>
</tr>
</tbody>
</table>
(II) Design Specification

The design specification provides more detailed information at component level. Developing new products without identifying the features of similar products may lead to loss in market share. Benchmarking is a process to show the features which are shortcomings of similar products so that design practitioners can draw up specifications to highlight the innovative parts of the products.

(a) Benchmarking

Benchmarking is essential to determine the relationship of new products to competing products. Innovation is essential to survival and generic benchmarking helps designers to gain an insight into the product trends through studying similar products or process, patents and published literature. Figure A.4 shows that data schema of product benchmarking contains the information of design problems and the findings of existing practice to solve the problems.

Figure A.4 Schema overview of product benchmarking
The description and data type of elements for benchmarking are listed in Table A.3.

Table A.3 Elements of product benchmarking

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Date of creation of the document</td>
<td>Date</td>
<td>Y</td>
</tr>
<tr>
<td>Project Name</td>
<td>The title of the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Current Practice</td>
<td>The existing practices including the related problems and published work in the field relevant to the proposed project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Search Result</td>
<td>The finding of quantitative information such as trade name and citation</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Category</td>
<td>Categories of search results</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Lead User</td>
<td>The major users of the product</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Experts</td>
<td>Person who has experience, skill in a particular knowledge domain</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Patents</td>
<td>The patent that was granted for the use of the invention</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Published literature</td>
<td>Related journals and magazines</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Similar product</td>
<td>Similar products that are used for benchmarking</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Comment</td>
<td>Comment that is related to customer expectations</td>
<td>String</td>
<td>N</td>
</tr>
</tbody>
</table>
(b) **Drawing up a product target design specification.**

A design specification is a summary that consolidates the findings of the potential market. A list of metrics includes the weighting for each product which indicates the importance of the customer needs. The design specification not only helps to document the design process, but also provides references to prevent design activities from deviating from the project mission stated in the stage of needs recognition. The preliminary design specification (See Figure A.5 and Table A.4) can deal with the component level of functional and physical performance as well as with the whole life-cycle issue by considering the perspectives of clients, designers and manufacturers.

![Figure A.5 Schema overview of design specification](image-url)
Table A.4 Elements of design specification

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Title</td>
<td>The title of the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Team name</td>
<td>The design team which is responsible for the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Purpose</td>
<td>The aims of the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>New or special feature</td>
<td>Unspoken or unexpected customers' need that can be a source of new/special features</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Competition</td>
<td>The competitive products and trade names of competitors</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Intended market</td>
<td>The potential market of the product</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Need for product</td>
<td>The expected, spoken, unspoken, and unexpected requirements which may be found in documents relating to customer expectation</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Relationship to existing product lines</td>
<td>The relationship of new product with the existing product line</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Market demand</td>
<td>Demand driven by the potential market</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Costs</td>
<td>Cost that relates to production and delivery, maintenance and disposal</td>
<td>Double</td>
<td>Y</td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
<td>Type</td>
<td>Required?</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------</td>
<td>-----------</td>
</tr>
<tr>
<td>Functional</td>
<td>Three functional requirements that capture the intended behavior of the system, the intended behaviors of the product and what the system will do. These behaviors may be expressed as functions, tasks, services that the system or product needs to perform. Therefore, functional requirements do not include performance characteristics, operating conditions, use cases, or specifications</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical requirements</td>
<td>Geometric entities that have meaning in the definition and manufacture of a product</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Service environment</td>
<td>The characteristics of the environment that make a product more readily serviceable.</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Life-cycle issues</td>
<td>The issues that relate to product life cycle including product introduction, growth, maturation and decline. Life-cycle issues are increasingly used by industries, governments and environmental groups to assist with decision making for environment-related strategies and materials selection</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Applicable codes and</td>
<td>A systematic collection of regulations and rules of procedure or conduct</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>standards</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table A.4 Elements of design specification (cont)

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human factors</td>
<td>Human factors refer to the characteristics of human beings that are applicable to the design of systems and devices of all kinds. It furthers serious consideration of knowledge about the assignment of appropriate functions for humans and machines, whether people serve as operators, maintainers, or users in the system. Also, it advocates systematic use of knowledge to achieve compatibility in the design of interactive systems of people, machines, and environments to ensure their effectiveness, safety, and ease of performance.</td>
<td>String</td>
<td>Y</td>
</tr>
</tbody>
</table>
| Corporate constraints   | 1. Constraints are used to fully define a model and to drive parametric or geometric systems. The algorithms used to work with constraints are known as constraint management  
2. Restrictions or boundaries that have impact on overall capability, priority, and resources | String | Y         |
| Comment                 | Comment that relates to customer expectation                                                                                                                                                                | String | N         |
(III) Conceptual Design

During conceptual design, concept generation is implemented prior to concept evaluation. A problem statement is defined as one that develops new features under corporate constraints where problem solving is the starting point of concept generation.

(a) Functional decomposition

A rational problem solving procedure should be conducted by analyzing the overall problem (function) and then breaking down the overall problem(function) into sub-problems (sub-functions). The data schema of functional decomposition (See Figure A.6 and Table A.5) is designed for capturing the information of the overall product function, main function and sub functions. Some comments about the functional decomposition method should be given to explain the relationship of functional decomposition.

![Figure A.6 Schema overview of functional decomposition](image)
Table A.5 Elements of functional decomposition

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>The date the document was created</td>
<td>Date</td>
<td>Y</td>
</tr>
<tr>
<td>Mechanical Design Team</td>
<td>Team members of the mechanical design team</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Project Name</td>
<td>The title of the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Overall Product Function</td>
<td>An abstracted description of work that a product must perform to meet customer needs</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>List of functions</td>
<td>A list that describes the main function and sub-functions</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Main function</td>
<td>A description of the main work that a product must perform to meet customer needs</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Sub-function</td>
<td>A description of other work that a product should perform to meet customer needs</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Comment</td>
<td>Comments that relate to customer expectations</td>
<td>String</td>
<td>N</td>
</tr>
<tr>
<td>Citation</td>
<td>A quoting of an authoritative source for substantiation</td>
<td>String</td>
<td>N</td>
</tr>
</tbody>
</table>

A sub-function is fulfilled by formulating the concept. This is done by adopting appropriate material or applying adequate mechanical design. Since patents may be given for some special material or mechanical design, designers may need to take it into consideration during concept generation. Designs for the sub-functions should be synthesized to form the overall function.
(b) Concept generation

The data schema of design concept generation includes a functional decomposition chart, the links of design concept sketch, concept description and the weighting of each concept description (See Figure A.7 and Table A.6).

Figure A.7 Schema overview of concept generation
<table>
<thead>
<tr>
<th><strong>Element Name</strong></th>
<th><strong>Description</strong></th>
<th><strong>Type</strong></th>
<th><strong>Required?</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>The date the document was created</td>
<td>Date</td>
<td>Y</td>
</tr>
<tr>
<td>Mechanical Design</td>
<td>Team members of the mechanical design team</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Representative Team</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Title Name</td>
<td>The title of the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Functional Description</td>
<td>A functional description chart illustrates the relationship between function-components</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Chart</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design Concept Sketch</td>
<td>A hasty drawing that describes the design concept</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Concept Description</td>
<td>An abstracted description of work that a product must perform to meet customer needs</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Concept</td>
<td>An idea for a new product or system that is represented in the form of a written description, a sketch, block diagram or simple model. A concept is the earliest representation of a new product or of alternative approaches to designing a new product.</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Functional Requirement</td>
<td>Functional requirements capture the intended behaviors of the system or product – i.e. what the system will do. This behavior may be expressed as functions, tasks, or services the system or product is required to perform. Therefore, functional requirements do not include performance characteristics, operating conditions, use cases, or specifications.</td>
<td>String</td>
<td>N</td>
</tr>
<tr>
<td>Rating scale</td>
<td>Uses a rating scale(1-5), where 5 is very good</td>
<td>Integer</td>
<td>N</td>
</tr>
</tbody>
</table>
(c) Concept evaluation

Design concept evaluation is performed after concept generation to ensure the concept is within the economic scope and is feasible to manufacture. Using a list of evaluation criteria scores need to be assigned to the design concepts so as to indicate how well each concept performs. The graphical representation of design concept evaluation is shown in Figure A.8 while the elements of design concept evaluations are listed out in Table A.7.

Figure A.8 Schema overview of concept evaluation
Table A.7 Elements of concept evaluation

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>The date the document is created</td>
<td>Date</td>
<td>Y</td>
</tr>
<tr>
<td>Mechanical Design</td>
<td>Names of the members of the mechanical design team</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Representative</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Title Name</td>
<td>The title of the project</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Weight Decision</td>
<td>A method to analyse the degree to which the product meets the customers’ needs</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Matrix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluation Criteria</td>
<td>Engineering characteristics and any additions that are identified during the functional analysis</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Weighting</td>
<td>The evaluation criteria weighting factors can be created using an objective tree or the relative importance scores from the QFD.</td>
<td>Integer</td>
<td>Y</td>
</tr>
<tr>
<td>Metric and Unit</td>
<td>Metrics from the analysis of competing solutions that is determined during functional analysis</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Marginal Metric</td>
<td>Value that indicates the customer needs are marginally fulfilled</td>
<td>Integer</td>
<td>Y</td>
</tr>
<tr>
<td>Value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ideal Metric Value</td>
<td>Value that indicates the customer needs are totally fulfilled</td>
<td>Integer</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concept</td>
<td>An idea for a new product or system that is represented in the form of a written description, a sketch, block diagram or simple model. A concept is the earliest representation of a new product or of alternative approaches to designing a new product</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
<td>Type</td>
<td>Required?</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-------</td>
<td>-----------</td>
</tr>
<tr>
<td>Metrics</td>
<td>The degree to which the product meets the customers’ needs</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Ranks</td>
<td>Uses a 0-10 scale (p. 189 of text) or other scheme to rank the performance of each design concept for each evaluation criteria.</td>
<td>Integer</td>
<td>Y</td>
</tr>
<tr>
<td>Score</td>
<td>Scores indicate how well each concept performs in relation to each evaluation criteria: 0 for inadequate performance, 1 for weak performance, 2 for satisfactory performance, 3 for good performance, and 4 for excellent performance.</td>
<td>Integer</td>
<td>Y</td>
</tr>
</tbody>
</table>
(IV) Detail Design

In the stage of detail design, the design concept is finalized and the detailed information of the product such as dimensions and tolerances of the fabricated components are marked in the engineering drawing.

(a) Cost Estimation

The cost of producing the manufactured product is estimated by aggregating the direct labor cost, direct material cost, tooling and equipment cost (See Figure A.9).

Figure A.9 Schema overview of cost sheet
Table A.8 lists out the cost components in cost sheet that helps engineers to find out the direct and indirect cost.

Table A.8 Elements of cost sheet

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part Number</td>
<td>The number that is assigned to the part</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Date</td>
<td>The date the document is created</td>
<td>Date</td>
<td>Y</td>
</tr>
<tr>
<td>Part Name</td>
<td>The description of the part</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Quantity used in final product</td>
<td>The quantities of the parts that are used to assemble the whole product</td>
<td>Integer</td>
<td>Y</td>
</tr>
<tr>
<td>Cost Table</td>
<td>A cost table is a multidimensional data base in which cost is captured for several levels of a number of attributes for either the parts or functions of a product. Cost tables are used to develop early estimates of the cost of a design based on product or part parameters or functions. Cost tables are primarily used by Japanese companies.</td>
<td>String</td>
<td>N</td>
</tr>
<tr>
<td>Operation Number</td>
<td>The number assigned to the operation for purposes of identification</td>
<td>Integer</td>
<td>Y</td>
</tr>
<tr>
<td>Operation Description</td>
<td>The detailed description of the manufacturing process</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Equipment</td>
<td>Cost of equipment used in product fabrication</td>
<td>Double</td>
<td>Y</td>
</tr>
<tr>
<td>Tooling</td>
<td>Cost of tooling that is used in product fabrication</td>
<td>Double</td>
<td>Y</td>
</tr>
<tr>
<td>Direct Labour</td>
<td>Labour cost that is directly chargeable to a product</td>
<td>Double</td>
<td>Y</td>
</tr>
</tbody>
</table>
Table A.8 Elements of cost sheet (cont)

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Material</td>
<td>Material cost that is directly related to a product</td>
<td>Double</td>
<td>Y</td>
</tr>
<tr>
<td>Equipment Available Cost</td>
<td>The cost of equipment that is available for production</td>
<td>Double</td>
<td>Y</td>
</tr>
<tr>
<td>Total Cost</td>
<td>Cost that summarizes direct labor, direct material, equipment, tooling and operation cost.</td>
<td>Double</td>
<td>Y</td>
</tr>
</tbody>
</table>
(b) Prototyping

Prototyping is transforming the abstract concept to physical 3-D model so that the actual functionality, form and features can be investigated using the prototype. There are several prototyping methods given in PIML schema which are (1) Plastic Model, (2) Plaster Duplicate (3) PU Duplicate (4) Clay Model (5) Wax Model and (6) Machining Model (See Figure A.10 and Table A.9).

Figure A.10 Schema overview of prototyping
Table A.9 Elements of prototyping

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Type</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Number</td>
<td>The number assigned to the part</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Document Number</td>
<td>The number assigned to this document</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Start Date</td>
<td>The date that the prototyping stage starts</td>
<td>Date</td>
<td>Y</td>
</tr>
<tr>
<td>Expected Completion Date</td>
<td>The date that the project finishes</td>
<td>Date</td>
<td>Y</td>
</tr>
<tr>
<td>Client</td>
<td>The name of the customer</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>Prototype Required</td>
<td>Various kinds of prototyping methods are provided</td>
<td>Choice</td>
<td>N</td>
</tr>
<tr>
<td>Plastic Model</td>
<td>A 3-d model made of plastic</td>
<td>Boolean</td>
<td>Y</td>
</tr>
<tr>
<td>Plaster Duplicate</td>
<td>A 3-d identical copy of plastic</td>
<td>Boolean</td>
<td>Y</td>
</tr>
<tr>
<td>PU Duplicate</td>
<td>A 3-d model made of PU</td>
<td>Boolean</td>
<td>Y</td>
</tr>
<tr>
<td>Clay Model</td>
<td>A 3-d model made of clay</td>
<td>Boolean</td>
<td>Y</td>
</tr>
<tr>
<td>Wax Model</td>
<td>A 3-d model made of wax</td>
<td>Boolean</td>
<td>Y</td>
</tr>
<tr>
<td>Machining Model</td>
<td>A 3-d model fabricated by a machining processes</td>
<td>Boolean</td>
<td>Y</td>
</tr>
<tr>
<td>Attached Information</td>
<td>Additional information for reference</td>
<td>String</td>
<td>N</td>
</tr>
<tr>
<td>Initiated by</td>
<td>The person who initiates the prototyping process</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Initiated date</td>
<td>The date the document is created</td>
<td>Date</td>
<td>Yes</td>
</tr>
<tr>
<td>Approved by</td>
<td>The person who approves the prototyping process</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Approved date</td>
<td>The date the document is approved</td>
<td>Date</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Appendix B

Product Development Survey

Objective: To find out the existing practice of the company during product development. It will investigate the comprehensiveness of the way in which information systems are deployed to assist knowledge workers during product development.

Survey of the existing practice of product development

1. What is the common problem(s) found in product development? (Can choose more than one)
   - □ Iterative product development
   - □ Short delivery time
   - □ Unclear customer requirement
   - □ Insufficient production capacity
   - □ Technological constraints
   - □ Others: ________________________

2. How does your company resolve the problem(s) during product development? (Can choose more than one)
   - □ By experience
   - □ Aid with computer system
   - □ Group Meeting
   - □ Adopt Supervisors’ Advice
   - □ Others: ________________________

3. How many design changes do your company usually encounter in product development process?
   - □ 1-5
   - □ 6-10
   - □ 11-20
   - □ More than 20

4. Do the same design changes occur again in other new product development processes?
   - □ Yes
   - □ No
5. Does your company realize that the loss of corporate knowledge is due to the resignation of experienced staff?
   □ Yes    □ No

6. Does your company have any method to manage knowledge?
   □ Yes Please Specify________________________
   □ No

System development for Product Lifecycle Management

7. How does your information system aid your work in your product development process?
   (Can choose more than one)
   □ Data Analysis    □ Decision support    □ Resolve the problem
   □ Computer Aid Design    □ Others:________________________

8. Does your company have an information system for Product Lifecycle Management (PLM)?
   □ Yes (Go to Q 9)    □ No (Go to Q15)

9. Is the PLM system developed by your company or bought from a software vendor?
   □ Self-developed (Go to Q10)    □ Off-the-Shelf (Go to Q13)

10. If the PLM system is developed by your company, did IT department develop it using object oriented technology?
    □ Yes (Go to Q 11)    □ No (Go to Q13)

---

1 Product Lifecycle Management as a concept within operation management is concerned with both the process of designing products and production systems, as well as with the realization of production, distribution, and service.
Appendix B

11. What aspect of Object Oriented (OO)² technology is used?
   □UML modeling □OO design and analysis □OO Programming

12. Is Object Oriented technology useful for software development?
   □Yes □No

13. Is artificial intelligence deployed in the Product Lifecycle Management (PLM) software?
   □Yes (Go to Q14) □No

14. Which kind(s) of artificial intelligence technique is (are) used? (Can choose more than one)
   □Case Based Reasoning □Rule-based reasoning □Fuzzy Logic
   □Neural Network □Others: ______________________

Product-Data Exchange during product development

15. How does your company exchange information with business partners? (Can choose more than one)
   □Fax □Tel □E-mail □EDI □XML □3rd party software
   □Others: ______________________

16. Has your company encountered data exchange difficulties among different system platforms?
   □Yes. Please specify: ________________________________
   □No

² Object-Oriented methods organize both the information, and the processing that manipulates that information, according to the real-world objects that the information describes.
17. How does your company resolve the problem of data exchange with business partners?

18. Please insert the appropriate number in the space available to rank each benefit of electronic data exchange method?
   4—most beneficial
   3—Average beneficial
   2—somewhat beneficial
   1—not beneficial
   □ Reduces typing errors □ Minimizes duplication of data entry
   □ Shortens data exchange time □ Maintains Data Integrity

The potential features of the information system

19. Does your company think an information system with adaptive³/self-learning features can increase efficiency?

   Strongly disagree Neutral Strongly agree
   1 2 3 4 5

20. Does your company think the information system with configurable methods to cope with the dynamic market is important?

   Strongly disagree Neutral Strongly agree
   1 2 3 4 5

³ Adaptation looks for prominent differences between the past case and the current case such that the information system adjusts the stored solution to fit for the needs of the current case.
21. Does your company think it is important that the information system should have data integration so that all relevant data can be automatically updated periodically?

<table>
<thead>
<tr>
<th>Strongly disagree</th>
<th>Neutral</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

22. Company Name: ________________________________

23. Number of employees in manufacturing sector:

☐ 0-35   ☐ 36-99   ☐ 100-299   ☐ 300-999   ☐ 1000 or above

24. Number of employees in other sectors

☐ 0-49   ☐ 50-99   ☐ 100 or above

25. Industry Sector

☐ Electronics & Electrical   ☐ Garments & Textiles   ☐ Gifts & Houseware

☐ Toys & Sporting Goods   ☐ Timepieces, Jewellery & Optical Goods

☐ Watch & Clocks   ☐ Others: _____________________________

26. Position in the company:

☐ Higher managerial or administrative   ☐ Supervisory

☐ Intermediate managerial or administrative   ☐ Professional

☐ Semi-skilled and unskilled manual worker   ☐ Skilled manual worker

☐ Clerical, junior administrative, trainee   ☐ Other: _____________________________

*************************************************************************** Thank You!***************************************************************************
Appendix C

The Measurement of Using Information System to Support Product Development

Survey of existing practice of product development

1. What are the common problems found in product development? (Can choose more than one)

![Common Problem found in Product Development](image)

Figure C.1 Common problem found in product development

2. How does your company resolve the problem(s) during product development? (Can choose more than one)

![Problem resolution Method](image)

Figure C.2 The method of problem solving adopted by companies
3. How many design changes do your company usually encounter in the product development process?

![Chart showing design changes](image)

Figure C.3 Design change encountered by companies

4. Do the same design changes occur again in other new product development processes?
   73 out of 95 answer "Yes"

5. Does your company realize that the loss of corporate knowledge is due to the resignation of experienced staff?
   68 out of 95 answer "Yes"

6. Does your company have any method to manage knowledge?

![Bar chart showing knowledge management](image)

Figure C.4 Knowledge management adopted by companies
System development for Product Lifecycle Management

7. How does your information system aid your work in the product development process?
   (Can choose more than one)

   Figure C.5 Function of information system

8. Does your company have an information system for Product Lifecycle Management (PLM)\(^1\)?
   18 out of 95 respondents have an information system for PLM.

9. Was the PLM system developed by your company or bought from a software vendor?

   Figure C.6 The proportion of PLM system self-developed or off-the shelf

\(^1\) Product Lifecycle Management as a concept within operation management is concerned with both the process of designing products and production systems, as well as with the realization of production, distribution, and service.
10. If the PLM system was developed by your company, did the IT department develop it using object oriented technology?
   14 out of 95 develop PLM system using object oriented technology

11. What aspect of Object Oriented (OO)\(^2\) technology is used?

   ![Pie chart showing aspects of OO deployment]

   Figure C.7 Aspect of OO deployment

12. Is Object Oriented technology useful for software development?
   11 out of 14 respondents realize that Object Oriented technology useful for software development.

13. Is artificial intelligence deployed in the Product Lifecycle Management (PLM) software?
   10 out of 95 respondents used AI in PLM software.

\(^2\) Object-Oriented methods organize both the information, and the processing that manipulates that information, according to the real-world objects that the information describes.
14. Which kind(s) of artificial intelligent technique is (are) used? (Can choose more than one)

![Bar Chart]

Figure C.8 The type of AI used by companies

Product Data Exchange during product development

15. How does your company exchange information with business partners? (Can choose more than one)

![Bar Chart]

Figure C.9 Method of information exchange with business partners
16. Has your company encountered data exchange difficulties among different system platform?

![Graph showing method of solving data exchange difficulties]

Figure C.10 Method of solving data exchange difficulties

17. How does your company resolve the problem of data exchange with business partners?

![Graph showing method of resolving the problem of data exchange]

Figure C.11 Method of resolving the problem of data exchange
18. Please insert the appropriate number in the space available to rank each benefit of the electronic data exchange method?
   4—most beneficial
   3—Average beneficial
   2—somewhat beneficial
   1—not beneficial

Table C.1 The benefits of data exchange

<table>
<thead>
<tr>
<th></th>
<th>Reducing typing error</th>
<th>Minimize duplication</th>
<th>Shortened data exchange time</th>
<th>Data integrity is maintained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>2.36</td>
<td>2.87</td>
<td>3.13</td>
<td>2.96</td>
</tr>
<tr>
<td>Rank</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>1.12</td>
<td>0.97</td>
<td>0.92</td>
<td>1.00</td>
</tr>
</tbody>
</table>

The potential features of an information system

19. Does your company think an information system with adaptive/self-learning features can increase the efficiency?
   Strongly disagree Neutral Strongly agree
   1 2 3 4 5

20. Does your company think the information system with configurable methods to cope with the dynamic market is important?
   Strongly disagree Neutral Strongly agree
   1 2 3 4 5

---

3 Adaptation looks for prominent differences between the past case and the current case such that the information system adjusts the stored solution to fit for the needs of the current case.
21. Does your company think it is important that the information system should have data integration so that all relevant data can be automatically updated periodically?

<table>
<thead>
<tr>
<th>Strongly disagree</th>
<th>Neutral</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Table C.2 Potential feature of information system

<table>
<thead>
<tr>
<th></th>
<th>Adaptive</th>
<th>Configurable</th>
<th>Automatic update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>3.49</td>
<td>3.66</td>
<td>3.66</td>
</tr>
<tr>
<td>Rank</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.85</td>
<td>0.87</td>
<td>1.15</td>
</tr>
</tbody>
</table>

Respondent’s Background Information

22. Company Name: __________________________

23. Number of employees in manufacturing sector:

24. Number of employee in other sectors

Table C.3 Number of employee in non-manufacture sectors

<table>
<thead>
<tr>
<th>Other\Mfg</th>
<th>0-35</th>
<th>36-99</th>
<th>100-299</th>
<th>300-999</th>
<th>&gt;1000</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-49</td>
<td>13</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>11</td>
<td>41</td>
</tr>
<tr>
<td>50-99</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>&gt;100</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>25</td>
<td>42</td>
</tr>
<tr>
<td>Total</td>
<td>23</td>
<td>8</td>
<td>7</td>
<td>16</td>
<td>41</td>
<td>95</td>
</tr>
</tbody>
</table>
25. Industry Sector

Figure C.12 Industrial sector of the respondents

26. Position in the company:

Figure C.13 Position of the respondents
**Appendix D**

**Documents from Case Companies**

---

**Csl Model No.:** MD8200 (PRC)

**Product Brief Description:** ENGLISH-CHINESE TALKING DICTIONARY WITHOUT SD CARD SLOT

---

**This Project Is**

- [X] Accepted
- [ ] Rejected

(This should be filled by Project Dept. Head after summarizing the result from individual Depts.)

---

**Estimated Engineering Sample Ready Date:** 12 July 01

**Estimated Pilot Run Date:** 24 Sep 2001

**Estimated First Shipment Date:** 19 Oct 2001

**Estimated Material Cost:** US$ 41

**Estimated Total Development Cost:** US$ 55350

---

**Remarks:**

ELECT.

---

**Mech. & Product Design**

**Elect. Pub.**

---

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECKED BY:</td>
<td>CHECKED BY:</td>
<td>APPROVED BY:</td>
</tr>
<tr>
<td>DATE:</td>
<td>DATE:</td>
<td>DATE:</td>
</tr>
<tr>
<td>17/6/01</td>
<td>2/3/01</td>
<td>27/6/01</td>
</tr>
</tbody>
</table>

---

**D1.**
GSL
Group Sales Ltd.

PROJECT REQUEST FORM:

PRF NO.: PRF:

PRF RE#: NO.:

GSL MODELING:

PRODUCT/INITIAL DESCRIPTION:

DOCUMENT/REVISION ATTACHED:

CUSTOMER NAME: 

TOTAL PAGES:

CUSTOMER PROD/REV REQUIREMENT NO.: 

REVISION:

SCHEDULED ENGINEER'S SAMPLE REQUEST DATE:

SCHEDULED RUN DATE:

SCHEDULED FIRST SHIPMENT DATE:

SCHEDULED FIRST SHIPMENT CFR:

MAK H/D/W/M MATERIAL COST:

OTHERS:

APPROVED BY:

REQUEST APPROVED BY:

DATE:

DATE:

D3.
Technical Analysis Report

1. Model No./Description:

1.1 Department: □ R&D (EE) □ R&D (MP) □ R&D (EP)

1.2 Estimated man power required: ___________ man month and engineering cost US$ ___________

1.3 Estimated product development time: ___________ months.

1.4 Estimated material cost: US$ ___________

1.5 Estimated new equipment and tooling investment in product development:

Cost: US$ ___________

1.6 Estimated new equipment and tooling investment in production:

Cost: US$ ___________

1.7 Estimated product development charge (Section 1.2 + 1.3): US$ ___________

Combined Information

2. Combined information: [R&D (EE) + R&D (MP) + R&D (EP)] (if applicable).

2.1 Total estimated product development charge: US$ ___________

2.2 Total estimated material cost: US$ ___________

2.3 Total estimated new equipment and tooling investment in production: US$ ___________

2.4 Combined product development schedule chart attached in sheet ___ of ___

PREPARED BY | CHECKED BY | APPROVED BY
--- | --- | ---
[Name] | [Name] | [Name]
[Signature] | [Signature] | [Signature]
CONFI METAL FINISHING CO., LTD.

WEIGHT X THICKNESS X KARAT = GOLD GRAM

1) ________________ x ________________ x ________________ = LOSS ________________ x ________________ x 25% =

2) ________________ x ________________ = ________________ = LOSS ________________ x ________________ x 25% =

3) ________________ x ________________ = ________________ = LOSS ________________ x ________________ x 25% =

4) ________________ x ________________ = ________________ = LOSS ________________ x ________________ x 25% =

5) ________________ x ________________ = ________________ = LOSS ________________ x ________________ x 25% =

6) ________________ x ________________ = ________________ = LOSS ________________ x ________________ x 25% =

1) TOTAL GOLD USED: __________________ 3) TOTAL GOLD USED: __________________

2) TOTAL GOLD USED: __________________ 4) TOTAL GOLD USED: __________________

(GOLD PRICE + HANDLE CHARGE) x PROFIT

1) ________________ x ________________ = ________________

2) ________________ x ________________ = ________________

3) ________________ x ________________ = ________________

4) ________________ x ________________ = ________________

5) ________________ x ________________ = ________________

6) ________________ x ________________ = ________________
### Appendix D

**Documents from Case Companies**

#### Project Acknowledgement Form

**GSL**
Group Sense Ltd.

**GSL Model No.**
MD6000 (FRC)

**Product Brief Description**
ENGLISH-CHINESE TALKING DICTIONARY WITHOUT SD CARD SLOT

---

**This Project is**
- [x] Accepted
- [ ] Rejected

(This should be filled by R&D(EE) Dept. Head after summarizing the result from individual Depts.)

**Estimated Engineering Sample Ready Date:**
12 JULY 01

**Estimated Pilot Run Date:**
24 SEP 2001

**Estimated First Shipment Date:**
19 OCT 2001

**Estimated Material Cost:**
US$ 41

**Estimated Total Development Cost:**
US$ 35350

**Remarks:**
ELECT.

---

**Mech. & Product Design**

**Elect. P.R.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Checked by:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approved by:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Signature:**

**Date:**
8/01

**Signature:**

**Date:**
7/8/01

**Signature:**

**Date:**
8/10/01
<table>
<thead>
<tr>
<th>PRODUCT BRIEF FORM</th>
<th>PBF NO.</th>
<th>PBF REV.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TO BE FILLED IN BY MAS DEPARTMENT REQUESTING FOR NEW PROJECT STUDY**

**CUSTOMER/DEPARTMENT NAME:**

**GSL MODEL NO. (if applicable):**

**PRODUCT BRIEF DESCRIPTION:**

**RESPONSIBLE R&D SECTION:**

- DOCUMENT SPECIFICATION ATTACHED
- TOTAL PAGES
- CUSTOMER PRODUCT REQUIREMENT DOC. NO.: ______________
- REVISION: ______________

**SCHEDULED ENGINEERING SAMPLE REQUEST DATE:** ______________

**SCHEDULED PILOT RUN DATE:** ______________

**SCHEDULED FIRST SHIPMENT DATE:** ______________

**MAX ALLOWABLE MATERIAL COST:** ______________

**COMMENT:**

**REQUEST INITIATED BY:**

**REQUEST APPROVED BY:**

<table>
<thead>
<tr>
<th>DEPARTMENT</th>
<th>SERVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPARTMENT</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Request Form</td>
<td>PRF No.</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>PBF Ref. No.</td>
</tr>
</tbody>
</table>

GSL Model No.

Product Spec. Description.

Document/Specification Attached

Customer Name [if any]:

Customer Product Requirement Doc. No.:

Revision:

Scheduled Engineering Sample Request Date:

Scheduled Pilot Run Date:

Scheduled First Shipment Date:

Scheduled First Shipment Qty.:

Max Allowable Material Cost:

Other:

Requested By:

Request Approved By:

Section Head or Designee

Signature

Date

Name

Position

Date

Designee

Position
1. Model No./Description: 

1.1 Department: ☐ R&D (EE) ☐ R&D (MP) ☐ R&D (EP)

1.2 Estimated man power required: _______________ man month and engineering cost US$ _______________

1.3 Estimated product development time: _______________ months.

1.4 Estimated material cost: US$ _______________

1.5 Estimated new equipment and tooling investment in product development:

____________________________

____________________________

Cost: US$ _______________

1.6 Estimated new equipment and tooling investment in production:

____________________________

____________________________

Cost: US$ _______________

1.7 Estimated product development charge (Section 1.2 + 1.5): US$ _______________

Combined Information

2. Combined information: [R&D (EE) + R&D (MP) + R&D (EP) (if applicable)].

2.1 Total estimated product development charge: US$ _______________

2.2 Total estimated material cost: US$ _______________

2.3 Total estimated new equipment and tooling investment in production: US$ _______________

2.4 Combined product development schedule chart attached in sheet ___ of ___.

<table>
<thead>
<tr>
<th>PREPARED BY</th>
<th>CHECKED BY</th>
<th>APPROVED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>SIGNATURE</td>
<td>NAME</td>
</tr>
<tr>
<td>DEPT/SECTION</td>
<td>DATE (DD/MM/YY)</td>
<td>DEPT/SECTION</td>
</tr>
</tbody>
</table>

D4.
<table>
<thead>
<tr>
<th>Department:</th>
<th>□ R&amp;D (EE)</th>
<th>□ R&amp;D (MP)</th>
<th>□ R&amp;D (EP)</th>
</tr>
</thead>
</table>

Model No./Description:

Product Comments:

**Appendix D**

D5.
WEIGHT X THICKNESS X KARAT X GOLD GRAM

1) ___________ x ___________ = ___________ x ___________ = ___________ x ___________ x 25% = ___________

2) ___________ x ___________ = ___________ x ___________ = ___________ x ___________ x 25% = ___________

3) ___________ x ___________ = ___________ x ___________ = ___________ x ___________ x 25% = ___________

4) ___________ x ___________ = ___________ x ___________ = ___________ x ___________ x 25% = ___________

5) ___________ x ___________ = ___________ x ___________ = ___________ x ___________ x 25% = ___________

6) ___________ x ___________ = ___________ x ___________ = ___________ x ___________ x 25% = ___________

1) TOTAL GOLD USED: ___________

2) TOTAL GOLD USED: ___________

3) TOTAL GOLD USED: ___________

4) TOTAL GOLD USED: ___________

(GOLD PRICE + HANDLE CHARGE) x PROFIT

1) (_________ + ___________) x ___________ = ___________

2) (_________ + ___________) x ___________ = ___________

3) (_________ + ___________) x ___________ = ___________

4) (_________ + ___________) x ___________ = ___________

5) (_________ + ___________) x ___________ = ___________

6) (_________ + ___________) x ___________ = ___________
Appendix E

Project Report for Gara Plastic Products Factory Limited

The Hong Kong Polytechnic University
Dept of Industrial and Systems Engineering
The Hong Kong Polytechnic
5267 University, Hung Hom,
Kowloon

Phone 2766 6584
Fax 2362

Project Report

Development of a knowledge based system to shorten product development lifecycle

Client Company: Gara Plastic Products Factory Limited

Supervised by: Dr. Henry C. W. Lau

Prepared by: Lee Ka Man
Rick Chan

Nov, 2002
Background of Toy Industry
GARA’s Best Opportunity for East Region Expansion

Introduction

Due to the keen competition, short product life cycle and changing market trends, manufacturers specialize in their own core value-added processes. Companies attempt a balance between being Original Equipment Manufacturers (OEM), i.e. suppliers who produce according to specific customer requirements, and their own brand of products. The firms with their own brand name prefer to take the strategy of product diversification but specialize in core business processes. OEM may therefore take the responsibility to complete their sub-contract processes on time and to finding ways to improve product quality by scrimping and saving on cost.

A paradigm shift of OEMs, which are changing from manufacturing with only the buyer’s product specification, to developing a complete product design with merely a design concept prior to production, is expected to continue into the next century. OEMs usually redesign the product by adding new product features to the old product. However, the existing capacity becomes the constraint if the existing equipments may not be able produce these new product features. Another challenge faced by the manufacturer is the shortness of the product development cycle. Delivery time, which is around 45-90 days on receipt of a letter of credit, depends on the complexity of the product and the availability of material.
Statistics of the Toy Industry

According to the findings of the Hong Kong Trade Development Council (HKTDC), Hong Kong produces a wide range of toys with particular strength in plastic toys, including dolls, dolls' houses and other accessories, toy figures, toy guns and gimmicks such as beauty kits and doctor's kits. *Gara is a typical plastic toy company with its headquarters in Hong Kong and produces products similar to those mentioned above.* Other major categories made by Hong Kong Toy Industry are electronic toys and games, radio/remote controlled toys, battery-operated toys and metal toys. Taken together with re-exports, Hong Kong is the world's largest toy exporter. Computer and video game interactive entertainment is expected to continue to be a major growth area, while technology allows the development of "intelligent toys". With the popularity of the cyber world, linking toys with the Internet is another new development. In the past, mass production helped to reduce the cost of production. Nowadays, the trend of mass customization gives the toys and customers a sense of uniqueness which may helps customers to identify the difference between the company and other competitors.

The latest official statistics provided by HKTDC show that the gross output of the toy industry reached HK$ 670 million in 1999, with a total of 317 manufacturing establishments hiring 1,686 workers as of September 2000. There are 204 plastic toys manufacturing establishments. Most of these establishments are small and medium enterprises (SMEs), usually employing less than 100 workers in Hong Kong.

To reduce operation costs and stay competitive, the majority of Hong Kong toy makers have set up production facilities offshore, mainly in the Chinese mainland and also a few in other low cost economies like Thailand, Malaysia, Indonesia and the Philippines. In the wake of this relocation trend, many toy companies in Hong Kong have been reclassified as import-export
establishments, thus contributing to the apparent decline in the number of toy makers locating in Hong Kong. At the end of 2000, there were 4,680 import-export establishments hiring 30,474 workers.

On the other hand, Hong Kong's role is shifting towards quality control, management, marketing, product design, and prototype manufacturing and production management. This skew towards higher value-added activities is helping Hong Kong toy makers to increase their competitive edge, while expanding production capacity through relocation. Hong Kong companies are renowned for their compliance with safety standards, regulations and codes of practice, enabling them to secure OEM orders from overseas industry giants, including Mattel, Hasbro, Disney, Bandai and Tomy, which advocate ethical sourcing practices. The value-added process of Gara is product design and the creative character winning OEM orders from Disney for over 19 years.

The toy industry has employed a wide range of manufacturing technologies; computer aided design and manufacturing systems (CAD/CAM) are commonly used. Many manufacturers have also earned the International Organization for Standardization (ISO) 9000 certification. Gara puts great effort into achieving international standards and Gara has got ISO 9001:2000 and ISO14000.

Yet competition has remained keen, especially from indigenous Chinese enterprises in open items. The rapid expansion of production capacity on the mainland by Hong Kong manufacturers also put much downward pressure on prices and profit margins. Orders received are smaller in lot size while shorter delivery lead time is generally required.
Performance of Hong Kong Toy Exports

After a 7% increase in 2000, Hong Kong’s exports of toys, as reflected in trade statistics, fell by 18% in 2001. While domestic exports showed 7% decrease, re-exports declined by 18% in 2001. In the process, exports to the US, which took up almost half of Hong Kong’s total exports, dropped by 19%, while exports to the EU were down by 17% due to weaker demand in the UK. Despite Japan’s domestic weakness, exports to that country posted healthy gains of 5% with a rising trend in reverse imports. The exports to the Chinese mainland and the ASEAN dropped by 39% in 2001 and by 12% in the first five months of 2002.

Structurally, the continuing trend of Hong Kong manufacturers producing in the mainland to shift from re-exports to offshore trade, with toys increasingly shipped directly from the Chinese mainland or transshipped via Hong Kong, has reduced trade going through Hong Kong. This serves to reduce the volume growth of cargoes going through Hong Kong, thereby depressing re-export growth. Further hurting matters with rising competition, overseas buyers have exerted intense price pressure on Hong Kong manufacturers.

Table E.1 Hong Kong’s exports of toys

<table>
<thead>
<tr>
<th></th>
<th>2000</th>
<th>2001</th>
<th>January - May 2002</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HK$ mn</td>
<td>Growth</td>
<td>HK$ mn</td>
</tr>
<tr>
<td>Domestic Exports</td>
<td>569</td>
<td>-9</td>
<td>529</td>
</tr>
<tr>
<td>Re-exports</td>
<td>90,909</td>
<td>+7</td>
<td>74,789</td>
</tr>
<tr>
<td>Total Exports</td>
<td>91,477</td>
<td>+7</td>
<td>75,318</td>
</tr>
</tbody>
</table>

*Source: Profile of Hong Kong Toy Industrial, Hong Kong Trade Development Council*

http://toys.tdctrade.com/
Table E.2 Hong Kong’s total exports of toys by major market

<table>
<thead>
<tr>
<th></th>
<th>2000</th>
<th>2001</th>
<th>January - May 2002</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Share %</td>
<td>Growth %</td>
<td>Share %</td>
</tr>
<tr>
<td>USA</td>
<td>48.4</td>
<td>+5</td>
<td>47.6</td>
</tr>
<tr>
<td>EU</td>
<td>25.1</td>
<td>+9</td>
<td>25.1</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>6.9</td>
<td>+8</td>
<td>7.0</td>
</tr>
<tr>
<td>Germany</td>
<td>6.2</td>
<td>+26</td>
<td>6.6</td>
</tr>
<tr>
<td>Italy</td>
<td>2.8</td>
<td>+3</td>
<td>2.6</td>
</tr>
<tr>
<td>Japan</td>
<td>7.0</td>
<td>+18</td>
<td>8.9</td>
</tr>
<tr>
<td>China</td>
<td>3.2</td>
<td>-6</td>
<td>2.4</td>
</tr>
<tr>
<td>ASEAN</td>
<td>1.8</td>
<td>+31</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Source: Profile of Hong Kong Toy Industrial, Hong Kong Trade Development Council

http://toys.tdctrade.com/

Table E.3 Hong Kong’s total exports of toys by major category

<table>
<thead>
<tr>
<th></th>
<th>2000</th>
<th>2001</th>
<th>January - May 2002</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Share %</td>
<td>Growth %</td>
<td>Share %</td>
</tr>
<tr>
<td>Dolls &amp; Parts</td>
<td>6.0</td>
<td>+2</td>
<td>6.3</td>
</tr>
<tr>
<td>Toys other than Dolls</td>
<td>60.4</td>
<td>+5</td>
<td>61.3</td>
</tr>
<tr>
<td>Games</td>
<td>16.2</td>
<td>+11</td>
<td>16.0</td>
</tr>
<tr>
<td>Carnival Articles &amp; Decorations</td>
<td>17.4</td>
<td>+12</td>
<td>16.5</td>
</tr>
</tbody>
</table>

Source: Census & Statistics Department, Hong Kong SAR Government

It is suggested that Gara can analyze the past three to five years sales volume. The sales volume can be measured in terms of category, export country, contractors in order to have further planning of marketing strategy. Gara should plot the curve to compare with the overall toy industrial statistics. If the curves follow the toy industrial data, it is a normal
situation and it implies the company trading environment is according to the economic situation of Hong Kong. However, if the curve does not parallel the performance of Hong Kong's toy industry, it is an abnormal situation. In such an abnormal situation, is the performance of the company better than the overall performance of the Hong Kong toy industry or is the performance worse than that of the Hong Kong toy industry as a whole. If it is better the company should identify their competitive strength and keep it up. If it is worse the company should reengineer the process and make improvements.

The company can forecast the sales volume according to the past 3-5 years data. Yet, the changing trading environment needs to be considered in conducting forecasting. The growing opportunities in domestic sales in the Chinese mainland resulting from China's WTO entry may affect Hong Kong toy manufacturers. On the other hand, some foreign companies may place orders directly with Mainland China. It is time to consider establishing point of sales in Mainland China.

**Industry Trends**

The increasing popularity of the Internet age impacts significantly on the toy industry in terms of the industry structure and the type of toys. Use of the Internet has been growing exponentially, and 50.3% of people aged over 10 have access to a computer and 43.3% were actively engaged in the Internet in 2001. According to the (2001) survey of information technology amongst household members, it is clear that an increasing amount of people's time will be occupied by the Internet and this will displace time that might have been devoted to play.
The trend of e-commerce for toy retailing is quite obvious in the US and to a lesser extent in Europe. Online sales of traditional toys surged to an estimated US$350 million in 1999 in the US from about US$50 million in 1998. It is projected that such sales of toys will increase to US$1.5 billion in three years. A number of traditional toy retailers have also joined the cyber market of toys, such as toyrus.com, KBToys.com and walmart.com, in addition to eToys.com and Amazon.com. With more newcomers entering the toy e-tailing business, the industry is expecting that there may be a consolidation among these online suppliers.

The application of e-commerce in the business to business level may be more relevant to most Hong Kong manufacturers. Overseas buyers will increasingly use the Internet to communicate with smaller suppliers in Europe and Asia. Such a development may imply even smaller orders and shorter delivery lead time for manufacturers.

Meanwhile, the toy industry remains highly concentrated, and there is a rising trend of merging activities of overseas toy companies both in the US and the EU, which may further raise their bargaining power in sourcing. There is also a noticeable trend towards greater concentration in the retail sector of Hong Kong major toy export markets. Consumers in the US, in particular, are bombarded with highly efficient discounter chains. The discounters, who work on low margins and wide varieties, have forced traditional retailers to squeeze prices and inventories, as they struggle to compete against large stores.

Apart from the technology dimension of new toys, shorter life cycles make the risk of product development increase. As the life cycle of successful products gets shorter and children become more sophisticated in their demands, manufacturers will be easily caught out by the changing trends. The impact on Hong Kong suppliers is thus not just the lowering of profit
Appends E

margins, but also the need to invest more in R & D and to develop their own design capabilities or more value-added edges which cannot be substituted easily by competitors.

The mass market depends on aggressive advertising campaigns or promotional tie-ins for success. Increasingly, toy manufacturers are entering partnerships with companies from other disciplines, especially fast-food chain stores, in their promotion campaigns. Toy makers are also increasingly entering into licensing deals with movie studios to make products featuring film characters.

The Internet prompts the paradigm shift of traditional business operation to e-commerce. E-commerce is also paves the way for toy industries to distribute their products over the world. Although e-commerce is a trend of doing business in western countries, SMEs in Hong Kong only promote products with the website, or to make better use of it. SMEs allow clients to place orders through internet. However, the transaction such as payment is still done by manual processes.

**Product Trends**

In all likelihood, interactive toys, or smart toys, will continue to be the mainstay. The development and success of robotic pets well illustrate the adoption of advanced robotic technologies into toys. Sony's Aibo, Silverlit-Tiger's i-cybie and Manley's Tekno have led robotic pets to hit store shelves. It is generally expected that the craze for robotic pets will extend into 2002, alongside an expansion in product varieties. But realism is increasingly important in developing smart toys, especially in categories such as dolls. Interestingly, it has been found that children prefer realistic-looking dolls to full-functioning robots.
Although video or electronic games have stormed the market in recent years, some parents are concerned about their possible negative impacts on children, such as a lack of social interaction and educational value. Such concern may stimulate the demand for quality traditional toys. A case in point is the comeback of board games. Human interaction is the driving force behind the renewed interest in board games. E.g. Monopoly is tie-in with Pokemon, in contrast to many computer games which can be played alone. This is particularly so after the terrorist attacks on the US, as consumers are becoming more family- and community-oriented. However, some board games also have electronic components. For example, a Harry Potter-related board game going to be released by Mattel will be equipped with casting stones in which computer chips are installed. This should enable them to be used with other Potter products.

Most toy design practitioners may find many interesting designs to please children. However, we should not forget who makes the final decision to buy the toy and pay for the product. Educational toys are also emerging as a key category, whether electronic, non-electronic or a combination of the two. In particular, a wider adoption and more creative use of hi-tech technologies in such toys tend to stimulate demand. While older kids with computer skills have come to expect electronic interactivity, pre-school children may get their first introduction to information technology via a growing range of electronic educational toys.

With the rising popularity of the cyber world, linking toys with the Internet is another continuing trend. The so called web-compatible toys enable players to interact with the designated websites. For example, Lego has teamed up with Microsoft and Universal Music International for the launch of its latest product range of toys and CD-ROMs, which are backed up by a website where children can learn about the universe and follow the story of a
group of characters. Another case in point is Furpsville, a US based toy company, which produces story books and plush alphabets to stimulate the interest in reading. Plans are in the pipeline to e-mail story lines to parents, so they can participate in the child’s learning cycle.

Although licensing based on movie/TV cartoon characters has no warranty for success, licensed products are still a major factor for the toy industry. 2002 should continue to be a promising year for wizards, monsters and prehistoric action toys. The Harry Potter figure is drawing much attention, particularly with the launch of the movie in late 2001. Some other movie characters, say featured in The Monster, Inc. and the Lord of the Rings, are expected to hit the market and provide an extra impetus to the licensing business. In another development, patriotic merchandise such as action figures of firemen and policemen have been selling well after the terrorist attacks. This revival will likely go much beyond 2001.

Apart from movie characters, Internet characters are also another new emerging trend for stationery or toys. A Korean Internet character named Mashimaro became the most popular character at the end of the 2001. It was propagated quickly through ICQ and e-mail.

http://www.mashimaro.co.kr/

**Challenge facing by Gara Plastic Products Factory Limited**

Due to the keen competition, short product life cycles and changing market trends, manufacturers have specialized in their own core value-added processes. Companies attempt a balance between being Original Equipment Manufacturers (OEM), which refers to suppliers who produce according to specified customer requirements, and in their own brand production. Gara, which is an OEM, faces the challenge stated before. Gara has worked with Disney for more than 19 years and has manufactured over 500 Disney characters. In addition, Gara have
produced Warner Brothers characters (over 100 collections) and Sanrio characters, as well as Power Rangers, VR Troopers, Spiderman, X-men, Dragon Ball Z, Smurfs, Garfield, Snoopy, and many other internationally well-known cartoon characters.

A paradigm shift of OEMs, which change from manufacturing with only the buyer’s product specification to developing a complete product design with merely a design concept prior to production, is expected to continue into the next century. They are known as Original Design Manufacturers (ODM). Gara provides customized designs for various clients to fulfill customer requirements and market trends. Similar to most OEMs, Gara usually redesign the product by adding new product feature to the old product. However, the existing capacity becomes a constraint if the existing equipment may not produce the new product features. Another challenge faced by the manufacturer is the short product development cycle. The short delivery time causes the engineers to make decisions based on their past experience and knowledge rather than on detailed planning and analysis to accomplish each order. Design agility and manufacturing flexibility are the critical success factors which help to resolve the problem mentioned before.

Product development becomes the core value added process and the existing workflow is shown below. The product development includes

1. Design

2. Model making

3. Tooling

E.12
In their Design process, Gara collects customer requirements. According to the capacity of the company, top management determines whether the factory accepts the order or not. If the correspondents approve, the designers prepare either engineering drawings or conceptual drawings for clients based on their expectation. When customers approve those drawing, the other stage, Model making, will be started. The design process is shown in Figure E.1.

Figure E.1 Design workflow
In the model making stage, Gara may realize the client's design as a prototype in various materials to give clients a feel for the finished product before commencing production. Project engineers will give feedback to the prototype maker if they receive any comment from the clients. The prototype is revised until the clients or project engineers think that it satisfies the customer.

Figure E.2 Prototype making workflow
Before mass production, tooling samples are produced for getting customer's approval of the design, moulding and color. Tooling is the last stage of product development and it is the critical stage as any mistakes may lead to further changes where those changes may add a considerable cost to the products. Therefore, tooling should be correct at the start.

![Flowchart](image)

Figure E.3 The tool making workflow
Although the process flow of product development is documented and achieved the ISO 9000 standard, there is no systematic method written down how design ideas are generated. The novel product design is usually developed by modifying some functions of the existing products sold in the market which may not be sorted systematically. Some information of the market product is not recorded so that it is difficult for designer to find out the relevant product information for the new customer requirement. The duplication of products may occur in the domestic or foreign market due to the lack of record and communication with other project engineers. Gara would like to shift parts of the job of product development to PRC staff in order to reduce the workload of HK staff. However, data decentralization causes work duplication (e.g. such particular product ideas for a novel customer requirement) which becomes the barriers of shortening product development time.

Secondly the product category is difficult to categorize since some categories are mixed and some products may belong to two categories. Another concern is how to define the categories as Gara has manufactured a large variety of products.

Thirdly, there is no database for product development processes and most documents are hard copies. Therefore, data retrieval is not easy and most useful information is memorized by knowledge workers. They use on their experience to complete the jobs.
Chapter 2

Objective of Study
Local manufacturing industry has transformed from labor intensive and low cost production to a knowledge-based and high value-added model. Having clarified the trend and background of toy industry, Gara's strength and uniqueness in the toy market will now be identified. Product development is the main value-added process. It starts from a design drawing proceeds to the creation of a prototype and finally to tool making. Information technology helps to reach precedent capabilities for complete component based design deployment. Nowadays, Gara has started to deploy information technology by setting up information infrastructure in order to facilitate the communication between the Hong Kong headquarters and production sites of PRC. E-mail, fax and telephone are the core communication means between the PRC to HK.

In this study, case based reasoning; XML and OLAP are suggested to enhance the efficiency and effectiveness of the processes during the product development lifecycle. To strengthen knowledge management in the product development process, Case Based Reasoning is suggested for handling poor structure and incomplete data in the initial stage of the product development process. In this report, Case Advisor 4.12 is used. It is a user friendly CBR system which can be downloaded from the website of Simon Fraser University. Apart from being user friendly, Case Advisor 4.12 allows users to access images, drawings, scanned documents which is the critical criteria for selecting Case Advisor 4.12 for Gara Plastic Products Factory Limited.
As there is an enormous amount of data involved in the product development lifecycle, end users need to analyze corporate data for the purpose of making better decisions. This is of paramount importance. OLAP technologies are essential for delivering this end-user value and are a critical component of broader information technology architecture. Fast, consistent response to end-user requests is critical to interactive, ad-hoc exploration, comparison and analysis of data, regardless of database size and complexity.

To exchange data among various systems, the new universal data exchange standard, XML, is proposed. XML is a markup language developed by World Wide Web consortium for documents containing structured information. As XML is supported by most commercial software applications, XML has become a data exchange standard and it acts as the interface between various applications.
Technology for Product Life-cycle Management
Most OEMs have the tendency to shift to ODM as clients provide OEM with simple or incomplete design concepts but ask them to complete product design, manufacture the product and deliver the finish goods to them on time. Product life-cycle management (PLM) is important. By definition, PLM spans the worlds of product development and manufacturing operations. Efficient product life-cycle management includes providing innovative design, resolving manufacture constraints and capturing knowledge in product development.

Online Analytical Processing
OLAP — online analytical processing — is the foundation for a range of essential business applications, including sales and marketing analysis, planning, budgeting, statutory consolidation, profitability analysis, balanced scorecard, performance measurement and data warehouse reporting. Although OLAP is neither a new nor an obscure concept, it is still not widely understood.

Many organizations know that they need OLAP-based solutions, but those tasked to select and implement them may be new to the area, or may have lost track of its rapid developments. Selecting the right OLAP product is hard, but it is very important.
In this project, Gara Plastic Products Factory LTD has collaborated with The Hong Kong Polytechnic University to make improvement on product development through Data Mining and Data Integration. The OLAP product chosen for this project is the Microsoft Analysis Service which is included in the Microsoft SQL Server 2000.

![OLAP & Data Mining](image)

**Figure E.4 Overview of OLAP**

The purpose of Microsoft SQL Server 2000 Analysis Services is to provide rapid analysis access to data warehouse data. To accomplish this purpose, Analysis Services creates multidimensional cubes from data in the data warehouse fact and dimension tables. Numerical measures are also summarized into preaggregated values during cube construction. The design of the data warehouse structure can affect how easily these cubes can be designed and constructed. Cubes are stored in multidimensional structures that are designed for rapid query response, combining preaggregated information with raw fact data to respond to a wide variety of queries. Cubes can contain data summarized, copied, or read directly from the data warehouse. OLAP cubes, dimensions, and partitions are processed to incorporate new or changed data from the data warehouse. The method of processing an OLAP object depends on
the object and type of change made to the data warehouse, such as real-time cubes to automatically synchronize cube data with changes in the underlying relational database.

Regarding the usage of OLAP in previous section, we could simply say it is mainly used for building decision-making models. Such as:

Queries: What were the sales last year?

Analysis: Why were the sales figures so low?

Build models: If we change the supplier, what will be the final result?

Plans: What items are going to change? How will they change?

**Advantages of OLAP in E-Commerce**

Shorter deciding period

More data to help in making comparisons

More detail of the data

Clear and continuous improvement

**Features of OLAP**

Depending on different situations and needs, OLAP allows the users to browse data on their own fields of interest. OLAP is built by multidimensional concepts. The data is stored in a cube. The cube consists of a fact table, dimensions and measures
Fact table: A fact table is a table in the relational data warehouse that stores the detailed values for measures.

Dimensions: A list of labels that can be used to cross-tabulate values from other dimensions.

Measures: A summarized numerical value used to monitor the business is called a measure.

Scenario of applying OLAP in Gara

There are many client tools compatible with the Analysis Services 2000; two kinds of client tools have been provided:

With Office 2000/XP, Excel offers a method for providing values to a PivotTable report: an OLAP cube. Using an OLAP cube, a PivotTable report can communicate with the PivotTable Service to retrieve values from an Analysis server.

To create an Excel PivotTable report based on an OLAP cube, we use the Excel PivotTable Report Wizard, which uses the Microsoft Query applications to define and create an OLAP query file. The OLAP query file provides all the information needed to connect to an OLAP cube. When Microsoft Query returns control to the PivotTable Report Wizard, the wizard uses the OLAP query file to connect to the cube. The figure below shows a toolbox which lists all the dimensions. We can drag those dimensions to the field at the table. These fields are: Page Fields, Column Fields, Row Fields and Data Items which are the same as the ones in the Analysis Service.
Figure E.5 OLAP Excel Pivot Table

Figure E.6 OLAP Excel chart
In addition, it can create an interactive dynamic chart to show the data. When we drag the dimension to the fields, the chart will modify immediately to show the result.

Figure E.7 OLAP query

**Procedure Overview**

Firstly, we need to select which data we need to analyze, and then design the dimensions in the OLAP cube. We use the data sources this project that include Microsoft Access and Microsoft SQL. We combine those sources and setup an automatic execute task using the Data Transformation Services which is included as a tool in the Microsoft SQL Server 2000.
Data Transformation Services (DTS) addresses a vital business need by providing a set of tools that lets us extract, transform, and consolidate data from disparate sources into a single or multiple destinations supported by DTS connectivity. By using DTS tools to graphically build DTS packages or by programming a package with the DTS object model, we have created a custom data movement solution tailored for the project.

Figure E.8 Data transformation Services
Case Based Reasoning

Case based reasoning helps the engineering departments to diagnose the design problem and allows users to solve the problem within a few minutes provided there is a precedent case stored in the case base. On the other hand, capturing the knowledge in failed processes and storing the information in the case base lets the design practitioners become aware of what kind of design may lead to production problems. Although CBR may not provide creative or innovative design, it helps designers to avoid the same design faults committed by previous designers and this helps to shorten the development time. CBR system is composed of four main components which are Case Base, Retrieval Algorithm, Case Reasoner and Adaptor.

![CBR Cycle Diagram](image)

Figure E.9 CBR Cycle
Case base--- The case base contains the facts that describe the problem areas and the past experience, which are saved in the form of cases. Each case is assigned an index number to avoid exhaustive computation by nearest-neighbor.

Retrieval Algorithm---The retrieval process involves feature identification, searching, initially matching and case selection. The system should first identify features, which come from the problem and critical description. Having searched the case with the direct index, index structure or general knowledge, the system starts to perform initial matching. Two main retrieval techniques are adopted in case based reasoning which are Nearest-Neighbor Retrieval (NNR) and Inductive Retrieval. The most suitable case is selected for modification for use in the current situation.

Case reasoner --- It is composed of case retriever and problem solver. Case retriever plays a role in searching the case base and identifying the most appropriate neighbor. The method to identify the cases similarity is to use nearest neighbor algorithm, induction and knowledge guided induction. All retrieved cases are presented to case reasoner for further examination. The case reasoner tries to solve the new problem by modifying the old solution to conform to the new situation.

Adaptor --- Case reasoner takes account of the difference between the retrieved case and the new problem such that adaptor can adapt the past case solution to provide a reasonable solution for the current problem. Adaptation can be generally done by deploying rule base, generic algorithms and by manual adaptation.
Scenario of applying CBR in Gara

Case based reasoning can assist design practitioners to capture critical information to develop their creative products. The most similar products that the design practitioners have designed before are retrieved and modifications to them are made. This can reduce the product design time. Developing a new version of the product design from scratch would take far too long. CAD drawings with useful descriptions of the past designs can simplify the retrieval process. Figure E.10 is an interface of Case Advisor 4.12 and it helps to retrieve the past design and opens with a CAD System. The user may simply input the product features required by the market trend such as small, light and portable. Case based reasoning system may search the related past design according to those descriptions and find out the solution for the designers.

For example, design practitioners would like to find out some past design about Mickey Mouse. The user simply inputs Mickey into the Problem Description and then the system will retrieve several questions related to the past cases. The user can then click the answer for the question and just click the case.
Figure E.10 The interface of Problem Resolution of Case ADVISOR 4.12
The selected case will be displayed at right side of the screen. The detailed information of the case such as case name, description and solution is shown.

Figure E.11 The interface of Problem Resolution of Case ADVISOR 4.12

When users click here for more, the image of the respective product is displayed.

Figure E.12 The interface of Problem Resolution of Case ADVISOR 4.12
**Procedure Overview**

This module named "Case ADVISOR Case Authoring" is used to edit the case base and decision trees. The case base is a collection of past problems that have been resolved. These cases can be recalled when trying to solve a similar new problem. Both the case attributes and the questions that are associated with the cases can be edited. Click Program File and select Case ADVISOR Case Authoring. The following screen is displayed.

![Case Authoring Interface](image)

*Figure E.13 The interface of Case Authoring of Case ADVISOR 4.12*
Click "File" and select "New" to open a new file domain in the above dialogue box. Input the name of the new domain in the first field and the path in the second field. The name of the domain can be different from the directory location. Users can create the folder in a specific directory or ask Case ADVISOR to create it.

![New Domain](image)

Figure E.14 The input interface for new domain

Creating Decision Tree Editor

A decision tree is a problem solving paradigm in which a user can follow a step by step sequence in a tree-like structure to resolve a problem. At each step the user will answer a question and possibly need to perform a diagnostic action. From the view menu, choose decision Trees. The decision Tree Editor Window will appear.

![Decision Tree Editor](image)

Figure E.15 The interface of Decision Tree Editor
Click "New Tree" button to create a new tree and enter "Scrap rate analysis" for the tree name.

Figure E.16 The interface of Edit Tree

Click on the "Edit Steps" button. The Decision Tree view will appear with the chosen decision tree displayed graphically. Each node in the decision tree can be selected; a right-mouse click will then display a context-sensitive menu.

Figure E.17 The interface for creating decision tree

Highlight "Scrap rate analysis" and then right click "Add Step".

Type "Compare the scarp rate with those under normal situation"
Right Click "Add Action". Type "Is the scrap rate increased by 15%?"

Right click "Add Answer". Type "Yes"

Right Click "Add Step". Type "Find out the cause of the high scrap rate"

Right click "Add Ques". Type "Does come from MFG process?"

Right click "Add Answer". Type "Yes".

Right click "Add Step". Type "Check out the process parameter"

---

Figure E.18 The interface for creating decision tree

Figure E.19 The interface for creating decision tree
Create New Case

New problems that have been solved can be saved in the case base. When a similar problem arises, this case can be retrieved in the search process and the solution can be applied to the new problem.

To add a new case:

1. Click on the button labeled new at the bottom of the Case Authoring screen. Alternatively, click the right mouse button in the Case List and select New from the pop-up menu. In either case, select the "New Case" option in the dialog that appears.

Figure E.20 The interface of Case Authoring
2 In the Case Name field, type the name of the new case. This field cannot be left blank. In this example, the case name is "High scrap rate".

3 In the "Description" field, type the description and/or symptoms of the case. This field can be left blank.

4 In the "Solution" field, describe the action that was taken to resolve the problem. The solution text can contain HTML tags, including links to pages on the World Wide Web. This field can be left blank.

5 Files, decision trees, and keywords can be associated with the case, but are not required. Options for a Case explain how to set these options.

6 To associate questions with a case, refer to Associating Questions with a Case.

7 Click on the "OK" button to save the current case. It will be added automatically to the Cases List and to the case base.

Figure E.21 The interface of Edit Case
Having clicked the "Case Option" button, Case Options Dialogue Box promoted out.

![Case Options Dialogue Box]

Select the decision tree and appropriate step within the tree.

User can attached image, document by pressing this button

User generates keywords from case name, description, solution, advanced keywords by ticking the appropriate checkbox

**Figure E.22** The interface of Case Options

Keywords are extracted from the case base so that they can be used later by the Problem Resolution module as queries to search for cases. The text-to-case conversion extracts keywords based on all of the cases in the case base. The default Case Authoring extracts key words based only on one case.

**Create New Question**

The method of creating questions is similar that for creating cases. Click on the button labeled new at the bottom of the Case Authoring screen. Alternatively, click the right mouse button in the Case List and select New from the pop-up menu. In either case, select the "New Question" option in the dialog that appears.

---

**E.37**
Associate Question to Case

Questions and answers are associated with cases to further describe or qualify a case. During Problem Resolution, questions are posed to the user to determine which cases are most likely to be relevant to the problem. When a question is answered, the scores of the cases associated with that answer increase. Similarly, the scores of cases which are associated with that question but not with that answer, decrease. When associating a question to a case, one or many answers and weights may be set. That is, one answer to a question may have a higher weight for a case and a second answer may have a lower weight for this case. The weight is used to determine the relative importance of a question to the case. The higher the weight, the more important the question is to the case and the more effect it will have on the calculation of the case score.

Figure E.23 The interface of Case Authoring
Having clicked the "Associate" button, the Associate Question dialogue box appears as below.

![Associate Question Interface](image1)

1. Highlight the solution
2. Select whether relate answer to
3. Adjust the importance weighting

Click "Selected cases"

**Figure E.24 The interface of Associate Question**

The following screen shows the case associate with the question.

![Case Authoring Interface](image2)

**Figure E.25 The interface of Case Authoring**
Please See the Appendix A for Case Authoring, Problem Resolution and Domain Manager of Case ADVISOR 4.12.

**XML for Product Data Management**

Data exchange is main concern in product data management where ISO 10303 STandard for the Exchange of Product (STEP) was proposed as the standard to allow interoperability between data from diverse CAD application software packages. However, the STEP philosophy, which is to exchange engineering data in form of central and neutral format, may encounter exchange difficulties in dispersed manufacturing environment. The recent distributed manufacturing environment prompts another new emerging technology, Extensible Mark up Language (XML), which is the universal format for structured documents and data on the web. Rezayat (2000) suggested applying XML to STEP as XML is able to handle large amount of dynamic content. Creation of Product Knowledge Markup Language based on key characteristic as a main enabler for defining the ontology is implemented as knowledge based product development system. Based on web-friendly and well-understood syntax of XML and the methodology of STEP/ISO 10303, a new paradigm for product data exchange namely Product Data Markup Language (PDML) is proposed to facilitate the integration and interoperability of business process. PDML defines a set of Application Transaction Sets for defining the data requirements to communities For purpose of achieving distributed collaborative design and manufacturability, Manufacturability Mark-up Language (MML) incorporating with CORBA provide a rapid manufacturing feasibility assessment tool which can be used at different design and planning stage.

The literature review so far indicates that whilst there exist a number of publications related to the approaches and methodologies to achieve efficient product data interchange, the
research concerns the deployment of artificial intelligence to leverage knowledge in product development. However, forming a real time data access of corporate database to knowledge base (case base) has not received the attention it deserves. This paper attempts to propose a data-knowledge integration system for integrating case based reasoning with object technology and the emerging XML data interchange standard so as to realize the cross platform database and knowledge base communication for supporting knowledge discovery. In this research, Product Information Markup Language (PIML) embracing a set of XML integrated to object technology to provide new paradigms of product data exchange standard is suggested in order to establish a linkage between case base and database, thereby enhancing the product development efficiency.

Markup Language is defined by ISO 8879. SGML has been the standard, vendor-independent way to maintain repositories of structured documentation for more than a decade but it is not well suited to serve documents over the web. Based on this limitation, XML allows users to define tags, provide arbitrary document structure and model the data structure. Those features of XML are not possessed by the only viable alternative--- HTML. XML supports a wide variety of applications and forms a good couple with Java since both of them can operate across different platforms. XML should be beneficial to a wide variety of diverse applications: authoring, browsing and content analysis. Although the initial focus is on serving structured documents over the web, it is not meant to narrowly define XML as it also acts as an interface between different applications and database systems.

XML has become the next generation of data exchange format due to the flexibility of its self-defined tag. However, most applications require a rigid schema which cannot be defined arbitrarily by various users. Otherwise, the data exchange standard may become
anarchic and it no longer conforms to a particular data structure. Document Type Definition (DTD), which is a piece of code that defines the allowable structure in XML document, is used to validate the XML document structure. XML parser, which is a software engine, performs the check on the data to make sure XML document conforms to the assigned DTD.

In order to allow programs and scripts to dynamically access and update the content, structure and style of documents, a platform- and language-neutral interface called Document Object Model (DOM) has been developed. The document can be further processed and the results of that processing can be incorporated back into the web page. Combining the HTML, style sheet and script to animate the webpages, Dynamic HTML is deployed to perform unique characteristic of object technology, polymorphism. The feature of polymorphism, which is the ability to response differently for the same message, can be encoded in script.

**Scenario of applying XML in Gara**

XML is a markup language proposed by the World Wide Web Consortium. Similar to HTML, it can go straight to be used on the Internet. XML is comprehensively used due to its compatibility with other SGML such as HTTP. XML supports a wide variety of applications and it allows users to perform (1) authoring, (2) browsing and (3) content analysis.

(1) Users can author XML by Notepad or some commercial XML software tools such as XMLSpy or XMetal. Those software provide an Integrated Development Environment for the eXtensible Markup Language that includes: XML editing & validation, Schema/DTD editing & validation and XSL editing & transformation. Users can export data by
Microsoft® Access 2002 or spreadsheet by Microsoft Excel into XML format. As a result, an elementary programmer can easily author the XML document.

Figure E.26 XML with XMLSpy
(2) Those data or documents in XML format can be used across any platform. For example, design practitioners use Mac PC to design products. At the same time, he/she can access information from the design drawing form in the intranet through the browser. The following figure shows the XML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<product_ID>
  <product_ID>0100</product_ID>
  <product_Name>Wardrobe</product_Name>
  <product_Category>Wood Furniture</product_Category>
  <price>200</price>
</product_ID>
```

Figure E.27 XML opened with web browser

Since XML is not good for presentation, Extensible Stylesheet Language Transformations (XSLT) has evolved from the early Extensible Stylesheet Language (XSL) standard. XSL specifies a language definition for XML data presentation and data transformations. Data presentation means displaying data in some format and/or medium. Presentation is about style. Data transformation means parsing an input XML document into a tree of nodes, and then converting the source tree into a result tree.
Transformation is about data exchange. After transformation, the page is represented in HTML as below.

![Image](image.png)

Figure E.28 HTML is formed by serializing the XML file with XSLT

(3) Design practitioners can access tremendous amounts of information from various sources. The main concern is how to extract critical and meaningful information to make decisions. The following examples shows that the data retrieved from SQL 2000 server is displayed as XML format via IIS such that it can be read and consumed by Excel 2002 in order to achieve data analysis. For example, Gara set up the web server in HK and published the XML data in the website via IIS. The Staff of Gara in PRC can type in the address "http://localhost/XML/PI.xml."
Figure E.29 Connecting to web server

Having connected to the web server and retrieved the data, Excel shows the data in the following spreadsheet.

Figure E.30 XML is opened with spreadsheet
The Design practitioners can click the “Data” menu and select “Pivot table and Pivot Chart Report” and conduct a simple analysis. A project engineer could find out what kind of product category has been bought from the market with the highest cost. He/She can drag Category to the left column with the name of the product next to it, then drag the price field to the left row (A3). In the total, the project engineer can find out the price of each product and the total price for each category.

Figure E.31 Data analysis with pivot table

**Procedure Overview**

Before authoring XML, database is to require to be created for product development. Based on the quality manual of Gara, 3 critical forms, Design Drawing Start Form, Prototype Start Form and Tooling Start Form are found where 3 tables are found in the product development database. To make it simple, the field of table is based on the field of
those forms. Computer knowledge workers can create the database in SQL 2000 server and export the table from the given CD through “Import and Export Data”.

A. Import the Database

Users should first create a database named “Gara” in Enterprise Manager of Microsoft SQL Server.

Figure E.32 Snapshot of data import

Click Next in the Data Transformation Service Import/Export Wizard. Import the database named Gara in the attached CD. Select Microsoft Access as Data Source.

Figure E.33 DTS Import Wizard

E.48
Select **Microsoft OLE DB Provider for SQL Server** in Destination Field and choose "Gara" the database.

![Destination for data import](image)

**Figure E.34 Destination for data import**

Select **Copy table(s) and view(s) from source databases.**

Select all tables (DDSF, Product, PSF, and TSF) from the lists.

![Source table selection](image)

**Figure E.35 Source table selection**
Click Run Immediately, then the data will be exported to the SQL server.

Figure E.36 Executing package for data import/export

B. Connect the ASP to the SQL server

The function of ASP is to allow users to input data to SQL server through the web. A user can simply create the ASP from Access 2002.

Figure E.37 SQL Server connection
Figure E.38 Data connection wizard
Design Drawing Start Form

Figure E.39 Dragging the data field to ASP

The user can simply drag the field from DDFS to the right side of the form. Then the ASP of the Design Drawing Form is completed. The ASP (ddsf.htm) can be saved in the Web server (c:/inetpub/wwwroot) such that users can directly enter the data to SQL through the ASP. The user can access and input the data through http://localhost/ddsf.htm.
C. Configure SQL XML support in IIS

Users can follow the following step to access SQL Server Using HTTP and export data as xml.

1. Configure SQL XML IIS support

   ![Image](image.png)
   
   Figure E.40 Snapshot of IIS

2. Set the New Virtual Directory and configure the setting in General, Security, Setting, Virtual Name, Data Source frame.

   ![Image](image2.png)
   
   Figure E.41 Setting the new virtual directory

   In this example, Windows Integrated Authentication is selected in the Security frame.

   In the Data Source frame, select the local as server and Northwind as database.

   In the Setting frame, select all option by clicking the radio button.

   In the Virtual Name frame, select Enter templates in the Virtual name box (it can be any user specified name). In the Type list, select template. Enter the path (for example, C:\inetpub\wwwroot\northwind\template, assuming there is a subdirectory template in the physical directory associated with the virtual directory; however the existence of the path is not checked). Click Save to save the virtual name.
3. Save the template file in the path
C:\Inetpub\wwwroot\[virtualdirectoryname]\[templatename]

(In this example, the folders for virtual directory and template name are not created. Therefore, the path is C:\Inetpub\wwwroot)

The following template file (MyTemplate.xml) includes SQL query and all the records in the Customer Table in the northwind database are retrieved.

```xml
<sql:query>
SELECT * FROM Orders FOR XML AUTO
</sql:query>
</ROOT>
```

4. To access data in SQL server through a browser, a user can type
http://localhost/[virtualdirectoryname]/[templatename]/[templatefilename].xml (in this example, the address is http://localhost/nwind/templates/MyTemplate.xml)

Figure E.42 Accessing data in SQL server through a browser
Summary

To summarize the report, the following points are raised for consideration:

- Having reviewed the statistical data about the toy industry, it is suggested that Gara should find out their position in the market place. Based on the statistics of the HK Toy exports and HK’s Total Exports of Toys by Major Market or by category, Gara can foresee the market trend and set up a strategic plan for the company.

- Based on the several options for technologies suggested in chapter 3, Gara should decide the way to go, taking into consideration how to equip the company with information technology to enhance their competitiveness.

- It may be a merit that Gara is already equipped with an IT infrastructure and therefore it is time to think about creating a database and setting up a system to enable data sharing between Hong Kong and PRC.

- Case based reasoning is good for product development. However, how to capture knowledge worker’s knowledge is the major barrier. As a result, implementing case based reasoning requires full support of top management. Otherwise, it won’t be successful.
REFERENCES


Cavalieri, S.; Maccarrone, P. and Pinto, R., Parametric vs. neural network models for the estimation of production costs: A case study in the automotive industry,

Available online 6 November 2003


http://www.info.gov.hk/censtatd/eng/hkstat/fas/it/penetration_index.html


http://xml.coverpages.org/pdx.html


References

The 1st International Conference on Product Development and Manufacturing

Technology, pp. 23-30


Vol. 7, No. 3, p. 22-27


Industry, Vol.54, No.1, p.51-67

House, B. (1991), Comparing the greats (comparing database management systems),
Data Based Advisor, pp.170-182

integrated design and manufacture strategy using an expert system approach,
Applications of Artificial Intelligence in Engineering, p 165-172


IPC, IPC-2571Generic Requirements fro Electronic manufacturing Supply Chain
Communication-Product Data eXchange (PDX), Nov 2001

Irgens, C.S. (1995) Design support based on projection of information across the
product-development life cycle by means of case-based reasoning, IEE
Proceedings of Science, Measurement and Technology, Vol.142, No.5,
pp.345 –349

Kansei structure on shoes using self-organizing neural networks, International

teams, Journal of Computer-Mediated Communication, Vol. 3, No. 4

be quality, Quality, Vol.14, No.2, p.39-48


foundations to applications. Springer: Hong Kong.

customer-oriented information system for product concept development,


Li, S. and Yang, Q. (2000) Activating Case-Based Reasoning with active databases


OMG (1996) Product Data management Enabler Request for Proposal


http://www.kbs.twi.tudelft.nl/Research/Projects/AIDA/


Multimedia: New York


http://www.tdctrade.com/econforum/tdc/tdc030702.htm

TDC Trade², Why Hong Kong as a business platform? A Survey of US Companies in the Pearl River Delta, 23 June, 2003,
http://www.tdctrade.com/econforum/tdc/tdc030602.htm


Usage Guide for the STEP PDM Schema V1.2, Release 4.3 Jan 2002

The STEP PDM Schema Homepage
http://www.pdm-if.org/pdm_schema/index.html


In Acta Philosophica Fennica, Fasc. XXI

W3C, (1999) XSL Transformation (XSLT)
http://www.w3.org/TR/xslt 16, November, 1999


http://www.w3.org/TR/xmlschema-1/ 2 May, 2001


