Steps towards computerized administration of factory information resources for CIM

This item was submitted to Loughborough University’s Institutional Repository by the/an author.

Additional Information:

• A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy at Loughborough University.

Metadata Record: https://dspace.lboro.ac.uk/2134/27145

Publisher: © Aiping Rui

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 2.5 Generic (CC BY-NC-ND 2.5) licence. Full details of this licence are available at: http://creativecommons.org/licenses/by-nc-nd/2.5/

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

Attribution-NonCommercial-NoDerivs 2.5

You are free:
- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
STEPS TOWARDS COMPUTERIZED ADMINISTRATION OF FACTORY INFORMATION RESOURCES FOR CIM

By
AIPING RUI

A Doctoral Thesis
Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy
of Loughborough University of Technology
March 1989

© by Aiping Rui, 1989
DEDICATION

Dedicated to my parents
for their love and support for my education
ACKNOWLEDGEMENTS

I wish to express my sincere thanks to my supervisors: Professor R.H. Weston and Mr A. Hodgson for their supervision, encouragement and help throughout this work.

I am grateful to the government of People's Republic of China and the British Council for providing financial support for my study.

I also take this opportunity to thank Mr D. Walters, Mr J.D. Gascoigne, Mr C.M. Sumpter, Mr I. Coutts, Mr G.P. Charles, Mr N.D. Carpenter, Mr S.T. Newman, Mr D. Round, Miss V. Launders, Mr R. Harrison and the staff of Computer Centre at L.U.T. for their technical assistance and valuable discussions, and to thank all the other staff of Department of Manufacturing Engineering for administrative work, in particular, Miss J.A. Howlett for her kindness. I appreciate the kindness from my office-mate Mr J. Kyrtsodious.

I sincerely thank my friends: Dr J. Pu, Mr G. Lu and Miss C. Sun for the friendship, inspiration and constant help I got throughout the period of my study, especially, during the period of the preparation of this thesis.

Thanks are also to many other friends of mine: Mr Z. Wu, Mr P. Zhang, Mr W. Wang, Mr J. Chan, Mrs H. Jiao, Mr J. Zhang, Mr X. Yan, Miss F. Zheng, Mr J. Qiao for overcoming difficulties and loneliness together during the period of my study. Finally, special thanks are to my parents and brothers, and my former teachers.
SYNOPSIS

KEY WORDS: CIM systems integration, flexible integration architectures, distributed databases, factory information administration, MAP/TOP networks, concurrency control, information standardization.

Being typical of manufacturing industry currently, "islands of automation" have severely limited further productivity increases. As being gradually realised, CIM (Computer Integrated Manufacturing) can provide opportunities for higher productivity and CIM systems integration is the major task for achieving CIM. With reference to the background, this research project was mainly concerned with formalising flexible CIM systems integration architectures and evolving generic and flexible integration tools and methods.

Based on an analysis of previous work concerning systems integrations and an analysis of the composition of contemporary stand-alone manufacturing entities and their interactions, a formal CIM systems integration architecture has been proposed. This proposal defines the need for three separate systems (or architectures) as follows:

(1) Digital Data Transfer System (DDTS),
(2) Information Administration System (IAS),
(3) Application Administration System (AAS).

With reference to the architectures proposed, the roles of MAP/TOP (Manufacturing Automation Protocol/Technical and Office Protocol) network standards and manufacturing information format standards have been assessed.
The focus of this thesis is on evolving generic and flexible integration tools and methods for building Factory IASs (FIASs). Database management, especially distributed database management, was identified as being an evolving technology suitable for creating a FIAS. Progress has been made in specifying requirements regarding the design and implementation of an efficient and reliable FIAS in the following aspects:

(1) the structural specification defining modular elements of a FIAS,

(2) the proposed use of manufacturing information standards for a FIAS to economically deal with the problem of "disparity in manufacturing information formats", i.e. SQL (Structured Query Language) as a standard database access language and EER (Extended Entity Relationship) data model as a standard data modelling method,

(3) the introduction of a novel concurrency control mechanism for a FIAS, termed a "multiple primary and improved two phase locking concurrency control mechanism",

(4) a prototype implementation of a generic and flexible FIAS, which integrates a relational and SQL-based DBMS, and a non-relational and non-SQL-based DBMS to facilitate concurrent compound information retrievals in SQL and concurrent simple information updates in SQL with access transparency and system configurability for flexible integration.

The formal and flexible CIM systems integration architectures and the methodologies evolved for building an efficient and reliable FIAS can be incorporated in both research projects and industrial implementations of CIM systems integration. At the same time, it should be stated that in a short term, the progress made in the aspect (3) is less important than in other aspects.
<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Figure Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Computerized elements of a CIM system</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>An overall CIM model</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>Direct and indirect benefits of CIM</td>
<td>15</td>
</tr>
<tr>
<td>2.4</td>
<td>Benefits achieved through interfacing and integration efforts</td>
<td>17</td>
</tr>
<tr>
<td>2.5</td>
<td>ISO/OSI reference model</td>
<td>20</td>
</tr>
<tr>
<td>2.6</td>
<td>MAP/TOP version 3.0 layer standards</td>
<td>22</td>
</tr>
<tr>
<td>2.7</td>
<td>Cost breakdown for a typical CIMAP cell</td>
<td>23</td>
</tr>
<tr>
<td>2.8</td>
<td>Approximate correspondence between SNA and ISO/OSI reference model</td>
<td>26</td>
</tr>
<tr>
<td>2.9</td>
<td>Typical transmitted Ethernet frame</td>
<td>26</td>
</tr>
<tr>
<td>2.10</td>
<td>Relation between ISO/OSI and X. 25 architecture</td>
<td>29</td>
</tr>
<tr>
<td>2.11</td>
<td>The format for a packet of information according to the KERMIT protocol</td>
<td>29</td>
</tr>
<tr>
<td>2.12</td>
<td>Relay types and their functions</td>
<td>30</td>
</tr>
<tr>
<td>2.13</td>
<td>Communication relays: a, repeater; b, bridge; c, router; d, gateway</td>
<td>31</td>
</tr>
<tr>
<td>2.14</td>
<td>Two approaches for information translation</td>
<td>33</td>
</tr>
<tr>
<td>2.15</td>
<td>Systems integration architecture</td>
<td>39</td>
</tr>
<tr>
<td>2.16</td>
<td>AMRF control hierarchy</td>
<td>41</td>
</tr>
<tr>
<td>Figure No.</td>
<td>Figure Title</td>
<td>Page No.</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>3.1</td>
<td>Facilities for integrating a PP&amp;C software package with a FAC controller</td>
<td>46</td>
</tr>
<tr>
<td>3.2</td>
<td>Example hierarchical PP&amp;C menus</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>An example of an operator’s communication with MICROSS</td>
<td>49</td>
</tr>
<tr>
<td>3.4</td>
<td>Specific integration tool</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>Functions of MICROSS PC System</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>Functions of MICROSS MRP System</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>An abstract model of an enterprise</td>
<td>63</td>
</tr>
<tr>
<td>4.4</td>
<td>Example precedence of PCB assembly events</td>
<td>65</td>
</tr>
<tr>
<td>5.1</td>
<td>Systems integration architecture</td>
<td>73</td>
</tr>
<tr>
<td>6.1</td>
<td>Lost update problem</td>
<td>88</td>
</tr>
<tr>
<td>6.2</td>
<td>Integrated information distribution optimization system</td>
<td>94</td>
</tr>
<tr>
<td>6.3</td>
<td>Hierarchical product realisation structure of an enterprise</td>
<td>96</td>
</tr>
<tr>
<td>6.4</td>
<td>Architecture of factory information administration system</td>
<td>104</td>
</tr>
<tr>
<td>7.1</td>
<td>Topologies of information transmissions</td>
<td>111</td>
</tr>
<tr>
<td>7.2</td>
<td>A subset of PP&amp;C ER diagram</td>
<td>126</td>
</tr>
<tr>
<td>7.3</td>
<td>Distributed database access</td>
<td>132</td>
</tr>
<tr>
<td>7.4</td>
<td>Appropriate stage for information translations</td>
<td>134</td>
</tr>
<tr>
<td>8.1</td>
<td>Lost update problem</td>
<td>138</td>
</tr>
<tr>
<td>8.2</td>
<td>Inconsistent retrieval problem</td>
<td>140</td>
</tr>
<tr>
<td>8.3</td>
<td>Inconsistent and incorrect copy problem</td>
<td>142</td>
</tr>
<tr>
<td>8.4</td>
<td>Example deadlock</td>
<td>146</td>
</tr>
<tr>
<td>8.5</td>
<td>Example cyclic restart</td>
<td>148</td>
</tr>
<tr>
<td>8.6</td>
<td>Example Multiversion timestamps</td>
<td>149</td>
</tr>
<tr>
<td>Figure No.</td>
<td>Figure Title</td>
<td>Page No.</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>8.7</td>
<td>lost update in PA method</td>
<td>154</td>
</tr>
<tr>
<td>8.8</td>
<td>Three phases of transaction execution</td>
<td>157</td>
</tr>
<tr>
<td>8.9</td>
<td>An example wait-for-graph</td>
<td>159</td>
</tr>
<tr>
<td>8.10</td>
<td>Efficiency evaluation of various concurrency control mechanisms</td>
<td>167</td>
</tr>
<tr>
<td>8.11</td>
<td>An example long waiting chain</td>
<td>169</td>
</tr>
<tr>
<td>8.12</td>
<td>Three transaction chain</td>
<td>171</td>
</tr>
<tr>
<td>8.13</td>
<td>Multiple transaction waiting case</td>
<td>173</td>
</tr>
<tr>
<td>8.14</td>
<td>Hierarchical deadlock detection scheme</td>
<td>174</td>
</tr>
<tr>
<td>9.1</td>
<td>Project facilities</td>
<td>179</td>
</tr>
<tr>
<td>9.2</td>
<td>Example information distribution &amp; duplication</td>
<td>183</td>
</tr>
<tr>
<td>9.3</td>
<td>Prototype factory information administration system</td>
<td>184</td>
</tr>
<tr>
<td>9.4(1)</td>
<td>Flow chart of system coordinator for multi-tasking operating system</td>
<td>186</td>
</tr>
<tr>
<td>9.4(2)</td>
<td>Flow chart of system coordinator for single tasking operating system</td>
<td>187</td>
</tr>
<tr>
<td>9.5</td>
<td>Flow chart for communication manager</td>
<td>188</td>
</tr>
<tr>
<td>9.6(1)</td>
<td>Flow chart for terminal server for multi-tasking operating system</td>
<td>190</td>
</tr>
<tr>
<td>9.6(2)</td>
<td>Flow chart for terminal server for single tasking operating system</td>
<td>191</td>
</tr>
<tr>
<td>9.7</td>
<td>Flow chart for message router</td>
<td>193</td>
</tr>
<tr>
<td>9.8</td>
<td>Flow chart for transaction decomposer</td>
<td>194</td>
</tr>
<tr>
<td>9.9</td>
<td>Flow chart for information assembler</td>
<td>195</td>
</tr>
<tr>
<td>9.10</td>
<td>Four schema concept</td>
<td>197</td>
</tr>
<tr>
<td>Figure No.</td>
<td>Figure Title</td>
<td>Page No.</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>9.11</td>
<td>Flow chart for schema manager</td>
<td>198</td>
</tr>
<tr>
<td>9.12</td>
<td>Flow chart for local DBMS front end</td>
<td>200</td>
</tr>
<tr>
<td>9.13</td>
<td>Architecture of SQL translator</td>
<td>202</td>
</tr>
<tr>
<td>9.14</td>
<td>Flow chart for embedded SQL precompiler</td>
<td>203</td>
</tr>
</tbody>
</table>
ABBREVIATIONS

2PL Two Phase Locking
AAS Application Administration System
AGV Automatic Guided Vehicles
AMRF Automated Manufacturing Research Facility
ANSI American National Standards Institute
ATE Automatic Testing Equipment
BDAS Basic Data Administration System
BOM Bill of Material
BR Boundary Representation
CAD Computer Aided Design
CAM Computer Aided Manufacturing
CAPP Computer Aided Process Planning
CIAM Computer Integrated and Automated Manufacturing
CIFM-85 Computer Integrated Flexible Manufacturing -1985
CIM Computer Integrated Manufacturing
CIMAP Computer Integrated MAP
CNC Computer Numerical Control
CSM Construct Solid Model
DBMS Database Management System
DDAS Distributed Data Administration System
DDBMS Distributed DataBase Management System
DDTS Digital Data Transfer System
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMIS</td>
<td>Dimensional Measuring Interface Specification</td>
</tr>
<tr>
<td>DNC</td>
<td>Direct Numerical Control</td>
</tr>
<tr>
<td>EDIF</td>
<td>Electronic Design Interchange Format</td>
</tr>
<tr>
<td>EER</td>
<td>Extended Entity Relationship</td>
</tr>
<tr>
<td>ESPRIT</td>
<td>European Strategic Programme for Research and Development in Information Technologies</td>
</tr>
<tr>
<td>FAC</td>
<td>Flexible Assembly Cell</td>
</tr>
<tr>
<td>FAS</td>
<td>Flexible Assembly System</td>
</tr>
<tr>
<td>FIAS</td>
<td>Factory Information Administration System</td>
</tr>
<tr>
<td>FMS</td>
<td>Flexible Manufacturing System</td>
</tr>
<tr>
<td>FTAM</td>
<td>File Transfer, Access and Management</td>
</tr>
<tr>
<td>IAS</td>
<td>Information Administration System</td>
</tr>
<tr>
<td>ICAM</td>
<td>Integrated Computer Aided Manufacturing</td>
</tr>
<tr>
<td>IGES</td>
<td>Initial Graphics Exchange Specification</td>
</tr>
<tr>
<td>IIDOS</td>
<td>Integrated Information Distribution Optimization System</td>
</tr>
<tr>
<td>IISS</td>
<td>Integrated Information Support System</td>
</tr>
<tr>
<td>IMDAS</td>
<td>Integrated Manufacturing Data Administration System</td>
</tr>
<tr>
<td>ISO</td>
<td>International Standards Organisation</td>
</tr>
<tr>
<td>IT</td>
<td>Information Translator</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MAP</td>
<td>Manufacturing Automation Protocol</td>
</tr>
<tr>
<td>MDAS</td>
<td>Master Data Administration System</td>
</tr>
<tr>
<td>ME</td>
<td>Manufacturing Entity</td>
</tr>
<tr>
<td>MHS</td>
<td>Message Handling System</td>
</tr>
<tr>
<td>MMS</td>
<td>Manufacturing Message Services</td>
</tr>
<tr>
<td>MRP</td>
<td>Material Requirements Planning</td>
</tr>
<tr>
<td>MRPII</td>
<td>Manufacturing Resources Planning</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MTBF</td>
<td>Mean Time Between Failures</td>
</tr>
<tr>
<td>NBS</td>
<td>National Bureau of Standards</td>
</tr>
<tr>
<td>NC</td>
<td>Numerical Control</td>
</tr>
<tr>
<td>OSI</td>
<td>Open System Interconnection</td>
</tr>
<tr>
<td>PA</td>
<td>Precedence Agreement</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>PDDI</td>
<td>Product Data Definition Interface</td>
</tr>
<tr>
<td>PDES</td>
<td>Product Data Exchange Specification</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logical Control</td>
</tr>
<tr>
<td>PP&amp;C</td>
<td>Production Planning and Control</td>
</tr>
<tr>
<td>QT</td>
<td>Query Translator</td>
</tr>
<tr>
<td>SAM</td>
<td>Semantics Associated Model</td>
</tr>
<tr>
<td>SET</td>
<td>Standard d’Echange et de Transfer</td>
</tr>
<tr>
<td>SNA</td>
<td>Systems Network Architecture</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>STEP</td>
<td>STandard for the Exchange of Product model data</td>
</tr>
<tr>
<td>T/O</td>
<td>Timestamp Ordering</td>
</tr>
<tr>
<td>TOP</td>
<td>Technical and Office Protocol</td>
</tr>
<tr>
<td>VMD</td>
<td>Virtual Manufacturing Device</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WFG</td>
<td>Wait For Graph</td>
</tr>
<tr>
<td>XBF</td>
<td>Experimental Boundary File</td>
</tr>
</tbody>
</table>
CONTENTS

THESIS TITLE
DECLARATION
DEDICATION
ACKNOWLEDGEMENTS
SYNOPSIS
LIST OF FIGURES
ABBREVIATIONS
CONTENTS

Page No.
1
II
III
IV
V
VII
XI
XIV

CHAPTER 1. INTRODUCTION

CHAPTER 2. LITERATURE SURVEY

2.1 Introduction

2.2 Computer Applications in Manufacturing

2.3 The Need for Automatic Information Integration

2.4 Computer Integrated Manufacturing (CIM) Concept

2.5 The Benefits of CIM Systems Integration

2.5.1 How to Justify Investments in CIM Systems Integration

2.5.2 Real Integration Benefits

2.6 Modern Integration Tools and Methods

2.6.1 Industrial Computer Networks

A. ISO/OSI MAP/TOP

B. SNA, Ethernet, X.25, etc.

- XIV -
B. SNA, Ethernet, X.25, etc.  
C. Interworking of Various Networks  

2.6.2 Manufacturing Information Format Standardization  
A. The Reason for Manufacturing Information Format Standards  
B. Major National/International Manufacturing Information Format Standards  

2.6.3 Systems Integration Architecture  

2.7 Summary  

CHAPTER 3. AN EARLY STAGE INTEGRATION ACTIVITY AND PROJECT SPECIFICATION  
3.1 Introduction  
3.2 Integrating a PP&C Package with a FAC Controller  
   3.2.1 Introduction  
   3.2.2 Facilities Utilised  
   3.2.3 Automating Remote Information Access  
3.3 Limitations of Pair Integration  
3.4 Project Specification  
3.5 Summary  

CHAPTER 4. ANALYSES OF MANUFACTURING ENTITIES AND THEIR INTERACTIONS FOR INTEGRATION  
4.1 Introduction  
4.2 Composition of Manufacturing Entities  
4.3 Interactions Between Manufacturing Entities  
4.4 Definition of CIM  
4.5 Summary
CHAPTER 5. SYSTEMS INTEGRATION ARCHITECTURE AND ASSESSMENT OF AVAILABLE INTEGRATION TOOLS AND METHODS

5.1 Introduction
5.2 Functional Requirements for Systems Integration Tools
5.3 Systems Integration Architecture
5.4 Assessment of Available Integration Tools and Methods
   5.4.1 Introduction
   5.4.2 The Role of ISO/OSI MAP/TOP
      A. FTAM
      B. MHS
      C. MMS
   5.4.3 The Role of Manufacturing Information Format Standards
   5.4.4 The Importance of Database Management Technology
5.5 Summary

Chapter 6. ARCHITECTURAL DESIGN OF A FACTORY INFORMATION ADMINISTRATION SYSTEM

6.1 Introduction
6.2 Importance of Distributed Database Management Technology for CIM Systems Integration
6.3 Issues of Distributed Database Management
   6.3.1 List of Issues
6.3.2 Data Models, Information Formats and Information Manipulation and Definition Languages
6.3.3 Information Distribution Optimization
6.3.4 Concurrency Control
6.3.5 Transaction Decomposition and Processing Optimization
6.3.6 Reliability and Crash Recovery
6.3.7 Information Integrity
6.3.8 Information Security
6.3.9 Distribution of Administration Functions

6.4 The Proposal of Integrated Information Distribution Optimization System

6.5 Characteristics of Manufacturing Environments
6.5.1 Hierarchical Product Realisation Structure from the viewpoint of organization
6.5.2 Non-hierarchical(Heterarchical) Product Realisation Structure from the Viewpoint of Information Processing

6.6 Architectural Design of Factory Information Administration System
6.6.1 Administration Approaches
6.6.2 The Proposal of Decentralization Favoured Administration
6.6.3 Modular Specification of Factory Information Administration System

6.7 Summary
CHAPTER 7. A PROPOSAL FOR A MANUFACTURING INFORMATION STANDARD

7.1 Introduction

7.2 Disparity in Manufacturing Information Formats

7.3 Rationale for Establishing a Manufacturing Information Standard

7.4 Current Situations and Principles for Developing a Manufacturing Information Standard

7.5 Suggestions Concerning a Manufacturing Information Standard

7.5.1 Historical Development of Data Models

7.5.2 Suggestion for a Standard Manufacturing Data Model

7.5.3 Suggestion for a Standard Manufacturing Information Manipulation and Definition Language

7.6 PP&C Data Modelling and Information Manipulation

7.6.1 Modelling PP&C Information in Chen's ER Model

7.6.2 Manipulating PP&C Information in SQL

7.7 The Appropriate Stage for Information Translations

7.7.1 Introduction

7.7.2 Network Communication Perspective

7.7.3 Distributed Databases Management Perspective

7.7.4 Direct Information Transfers between CAD/CAM Databases

7.7.5 Summary of Perspectives
CHAPTER 8. ADVANCES IN CONCURRENCY CONTROL FOR DISTRIBUTED DATABASES

8.1 Introduction

8.2 Problems Without Concurrency Control Mechanisms
   8.2.1 Introduction
   8.2.2 Lost Update Problem
   8.2.3 Inconsistent Retrieval Problem
   8.2.4 The Incorrect and Inconsistent Copy problem

8.3 Review of Various Concurrency Control Mechanisms
   8.3.1 Introduction
   8.3.2 Basic Timestamp Ordering (T/O)
   8.3.3 Multiversion T/O
   8.3.4 Conservative T/O
   8.3.5 Multidimensional T/O
   8.3.6 Precedence Agreement (PA) Method
   8.3.7 Majority Consensus Approach
   8.3.8 Optimistic Approach
   8.3.9 Two Phase Locking (2PL)
   8.3.10 Timestamp Based Deadlock Prevention
   8.3.11 Unified Concurrency Control

8.4 Efficiency Evaluation of Various Concurrency Control Mechanisms

8.5 Multiple Primary and Improved Two Phase Locking Concurrency Control Mechanism

8.6 Summary
CHAPTER 9. A PROTOTYPE FACTORY INFORMATION ADMINISTRATION SYSTEM

9.1 Introduction
9.2 Consideration of Project Facilities
9.3 Functional Requirements of the Prototype System
9.4 System Architecture and Mechanisms
  9.4.1 System Coordinator
  9.4.2 Communication Manager
  9.4.3 Terminal Server
  9.4.4 Message Router
  9.4.5 Transaction Decomposer
  9.4.6 Information Assembler
  9.4.7 Schema Manager
  9.4.8 Local DBMS Front End
  9.4.9 SQL Translator
  9.4.10 Embedded SQL Precompiler
9.5 Operational Procedures
  9.5.1 Schema Specification and Message Header Format
  9.5.2 Example Compound Retrievals
  9.5.3 An Example Embedded SQL Program
  9.5.4 Example Simple Updates
  9.5.5 Example Updates with Information Duplication
9.6 System Configuration
  9.6.1 System Configuration for Introduction of New Information Tables
9.6.2 System Configuration for Transparency under Changes 219
9.7 System Implementation Mode 219
9.8 Summary 221

CHAPTER 10 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK 222
10.1 Research Findings and Conclusions 222
10.1.1 Introduction 222
10.1.2 Systems Integration Architecture 223
10.1.3 Design and Prototype Implementation of a FIAS 227
10.2 Recommendations for Future Work 232

REFERENCES 234

APPENDIX A. SELECTED FUNCTIONAL DESCRIPTION OF KERMIT FILE TRANSFER 249

APPENDIX B. UTILISED REPORT DETAILS OF MICROSS PP&C PACKAGE 251

APPENDIX C. AUTHOR’S PUBLICATIONS 260
CHAPTER 1
INTRODUCTION

Computers have found widespread industrial applications in performing design, management and control functions to enable significant improvements in product realisation. However, computerised manufacturing systems have been designed as stand-alone entities which do not interact effectively with each other. This fact has severely limited opportunities for further productivity increases in terms of reducing product lead time, bringing down product cost and increasing responsiveness to market changes. In other words, the success of Factory 2000 will be intimately linked to the degree of flexible integration achieved. Hence, in the author's opinion, systems integration is the major task for realising CIM (Computer Integrated Manufacturing).

Potential benefits of CIM have motivated the manufacturing community world-wide to invest in this field of research. For example, the General Motors and the Boeing corporation, since 1979, have injected momentum into the CIM systems integration movement by initiating and later jointly developing the ISO/OSI seven layer reference model based MAP/TOP (Manufacturing Automation Protocol/ Technical and Office Protocol) specifications for networking enterprise computers and computer-controlled machines. The overwhelming support from both users and vendors indicates that MAP/TOP based networks or derivatives of them will become commonplace in industry. Furthermore, significant funding from ESPRIT (European Strategic Programme for Research and Development in Information Technologies), United States and Japanese governments has been directed towards modelling and implementing
integrated manufacturing systems. Nevertheless, CIM systems integration is a relatively new research area so that the techniques evolved cannot be considered to be generic and conclusive.

Within the context of world-wide CIM systems integration research, along with the successful integration of a PP&C software package with a FAC controller in the early stages of this study, the aims of this research study are set as follows:

(1) to formalise the design of a flexible and generic architecture for CIM systems integration, which could have long term implications for CIM systems integration,

(2) to assess the roles of MAP/TOP standards and manufacturing information format standards, which is important for providing knowledge to potential users of those standards and for possibly influencing the future development of those standards,

(3) to develop generic integration tools and methods, which are what world-wide systems integrators are searching for.

The fundamental characteristics of contemporary or stand-alone manufacturing entities and their interactions are first analysed so that the overall systems integration architecture, which is based on the "three architecture" approach (namely, network architecture, information architecture, application architecture), can be formalised through substantiation, enhancement and improvement. The primary aim of this exercise is to formally segment the complicated integration problem into manageable sub-problems for creating a long term integration infrastructure. After the systems integration architecture is formalised, the role of MAP/TOP standards and manufacturing information format standards is assessed. Under the systems integration architecture determined, one of these sub-problems
is found to be the creation of a Factory Information Administration System (FIAS) which can be realised by harnessing database management technology. Subsequently, the thesis is focused on designing an efficient, reliable and cost-effective FIAS for providing enterprise-wide information services. A suitable architecture and modular specification of a FIAS are proposed through the review of contemporary database management technology and logical reasoning. Proposals are also made for using a standard manufacturing database access language and a standard manufacturing data modelling method. A significant problem area in distributed databases, on which a FIAS is based, relates to concurrency control. An efficient and reliable concurrency control mechanism is suggested based upon qualitative evaluation and comparison of modern concurrency control mechanisms. Finally, the implementation of a prototype heterogeneous FIAS is described.
CHAPTER 2
LITERATURE SURVEY

2.1 Introduction

The background concepts and major study themes of this research project will be introduced through a short review of common manufacturing application areas of computers, followed by an identification of the need for automatic information integration, a description of emerging CIM concepts and definitions. Then, discussions will be presented relating to a possible way to justify investment in CIM systems integration and the real benefits achieved in some of the major small scale and specific integrated manufacturing systems world-wide. In the latter part of this chapter, modern systems integration tools and methods will be reviewed and categorized under the following three headings:

(1) industrial computer networks,
(2) manufacturing information format standardization,
(3) systems integration architecture.

This chapter will be concluded by summarising the problems which must be solved before CIM systems integration can be realised.

2.2 Computer Applications in Manufacturing

Computers have found widespread applications in manufacturing industry, including product design, finance, production planning and control, manufacturing and assembly process control, quality control, etc.
In an office environment, computers have been used in many ways. Computer Aided Design (CAD) has been developed for two decades [CON69] [CON72] [CON74] [CON87] [BES83] and is widely used in companies of various sizes as well as in many research institutions, which have been improving efficiency and quality of product design considerably [CON80] [HAW88]. The scale of computer applications in business and production management (which mainly consist of finance, marketing, master production scheduling, capacity planning, material requirement planning and production scheduling) is by no means smaller than those in CAD. This observation is reflected in the literatures [WIL84] [HAR74] [HAR78] [LOC83] [PRA86] [SMO82] [LAW84] and the availability of commercial software packages [SYS87]. Commercial production planning and control packages offered by different software vendors vary in their function and scope [SYS87]. It is evident that master production scheduling, Material Requirements Planning (MRP) or Manufacturing Resource Planning (MRP II), production scheduling, stock control, and work-in-progress monitoring are the major software packages in the marketplace [SYS87].

On a shop floor, many types of computer controlled manufacturing machine exist, such as robots [RAT85] [ENG80], NCs [PRE77], CNCs [PUS83] [BOW84], PLCs [JOH85] [DOR83], material transportation and storage systems [MUL83] [CON86]. Manufacturing and assembly process planning, NC/CNC, robot and other machine programming, and quality control are all computer assisted. These are usually covered by the term CAM (Computer Aided Manufacturing) [CON78] [GRO84]. Since 1980, the availability of FMSs (Flexible Manufacturing Systems) has heralded a new era of discrete parts manufacturing by realizing production flexibility [BES86] [KUS86]. It must be stated that the definition for CAM sometimes is meant to cover NCs, CNCs,
DNCs, robots and other shop floor machines, FMS/FAS (Flexible Assembly System), quality control, process planning, and production management functions [GRO87]. Nevertheless, the varying scopes of those terms used by different people do not change the fact that computers have pervaded every corner of various manufacturing industries.

The efficiency, effectiveness and quality of product design, production planning and control, manufacturing and assembly process control, and other production activities have been substantially improved [HUB85] [PRE84] as the result of the use of computers. For example, CAD increases a designer's productivity in various ways [ENC83] [HOR86] [HAW88], which include:

1. faster creation of new designs of higher quality,
2. faster modification of existing designs at a lower error rate,
3. automatic generation of detail drawings and bill of material files.

By using computers, the speed and quality of management decision-making in activities, such as MRP, MRPII, scheduling, and manufacturing and assembly process control can be significantly improved [HAR78]. In general, it can be concluded that applications of computers in manufacturing have greatly increased productivity through reducing product cost and product lead time, and quickening the response to market changes [PRE84] [ING85].

2.3 The Need for Automatic Information Integration

The level of both electronic and mechanical automation has been significantly increased as the result of computer applications, and our society stands today in the dawn of an information age [HES85]. Toffler described "the information age" in general as "the third wave" [TOF80]. Many organisations other than
manufacturing companies have no products except information, and the value of their information assets is substantially greater than the value of their tangible assets. Good examples of this are the pharmaceuticals company Glaxo, the advertising agency Saatchi & Saatchi and the information distribution company Reuters [BEA88]. Information is also considered to be one of the most valuable assets in manufacturing industry [HES85] [DTI87] [BEA88].

Information in manufacturing forms a cycle from marketing and order entry, design, production planning and control, manufacturing and assembly process control, sensory feedback, and work in progress updating to product shipment to customers [GRO87] [RAN86]. In other words, information generated by one Manufacturing Entity (ME) (*) is needed at a number of other MEs [RAN86] [HES85]. Ranky uses the phrase 'almost everything relates to everything' in CIM to emphasise the close links among CIM subsystems [RAN86]. For example, process routings produced by a CAPP (Computer Aided Process Planning) system must be passed to all manufacturing and assembly cells to make parts [REM85]. Product geometric models from CAD are utilized by a CAPP to plan process routings, by CAM to prepare NC/CNC and robot programs and by computer aided inspection stations to measure product dimensional quality [HES85] [ING85]. The above two examples are "downward information flows". An example of "upward information flow" could be that a shop quality control manager wants to collect all quality information from all

(*) Manufacturing Entity (ME) can be defined as any manufacturing sub-system which contributes to product realization in the way of making decisions, physically transforming the states of components, or supporting the former two.
computer aided inspection stations [HES85]. All such information flow activities are usually achieved manually in conventional manufacturing environments. Such manual information communication links are notoriously slow, expensive and error-prone [HES85].

Realising that information is the key resource, the acquisition and effective use of the information will be the key to the success of any enterprise [BEA88]. The lack of electronic digital links between "islands of automation" is one of the major reasons for the following problems [ING85] [HES85]:
(1) enormous stock levels which tie heavy investment,
(2) long overall lead times,
(3) slow response to the market changes,
(4) high costs when implementing product design changes,
(5) low machine utilisation,
(6) poor quality of information both in timeliness and inaccuracy, and high costs associated with the management of information.

Although the widespread use of computers in manufacturing as stand-alone entities has increased productivity, it does not provide technology advantages over market competitors [SHR85]. The need to improve product quality and reliability, and to reduce product cost still remains [SHR85] [ING85]. Thus, there exists a great need for automatic information integration. Professionals in manufacturing business predict that the overall trend in engineering and manufacturing between now and the year 2000 is towards the development and implementation of CIM [SHR85] [ING85] (This concept will be elaborated in the next section). CIM will promise higher productivity and better competitiveness for those companies who are committing themselves to investing
In a long term, \( \checkmark \) large or medium \( \checkmark \) in CIM [SHR85] [ING85]. CIM is not optional for a company that intends to remain in business [SHR85].

2.4 CIM (Computer Integrated Manufacturing) Concept

Based on the recognition of the need for automatic information integration, various levels of understanding of modern manufacturing technologies and predictions of their development, a variety of definitions or explanations for CIM have been proposed.

In the Ingersoll Engineers’ report called Integrated Manufacturing [ING85], published definitions of CIM were listed. They are:

(1) CIM is an information structure supporting the free flow of all information resident in the system to any part of the system as needed.

(2) CIM is an amalgam of: computer-aided engineering and drafting, computer-aided manufacturing engineering, flexible manufacturing systems, tooling and quality support systems, in-process gauging and automated final inspection, automated storage and material handling, and operations control within a business system.

(3) CIM is a unified network of computing systems performing and/or controlling the totally integrated functions of business.

(4) CIM is the phased implementation of the integration of automated and non-automated systems into the manufacture of a product. It is an integration to the maximum level of beneficial usage, to serve both long-term profitability and quality of product.

(5) CIM is the organization of the company so that computer hardware and
software capabilities can be used to permit large organizations working on complex products to achieve the responsiveness of small organizations working on simple products.

(6) CIM is a strategy consisting of physical components and conceptual methodology to integrate the components.

Among the six definitions of CIM listed above, definitions 1-3 are more technical than the last three. The first definition emphasises the occurrence of automatic information flow whereas the second details the various manufacturing sub-systems which form CIM, and the third highlights the integration of business functions. However, the last three definitions describe CIM informally from the viewpoint of business and implementation.

In developing our discussion on CIM, let us consider three other, arguably, more comprehensive definitions or explanations for CIM. Groover [GRO87], Bunce [BUN85] and Ranky [RAN86] agree on a more general definition in that CIM is the integration of activities at all levels in a company to produce products.

Groover [GRO87] accepts that sometimes CAD/CAM is used interchangeably with CIM, but rather defines CIM as the application of computer technology to all of the operational functions and information processing functions in manufacturing from order receipt, through design and production, to product shipment, which has a broader scope than CAD/CAM by incorporating business functions as illustrated by figure 2.1. CAD and CAM are concerned principally with the engineering functions in design and manufacturing respectively. Groover uses CAM in a broader sense to encompass all manufacturing functions other than CAD and business functions such as financial accounting.
Figure 2.1 Computerized elements of a CIM system

(A COPY FROM [GRO87])
Bunce's definition of CIM [BUN85] is as follows: "a series of interrelated activities and operations involving the design, materials selection, planning, production, quality assurance, management and marketing of discrete consumer and durable goods... CIM is the deliberate integration of automated systems into processes of producing products... CIM can be considered as the logical organization of individual engineering, production and marketing/support functions into a computer integrated system".

Ranky [RAN86] points out that CIM is concerned with providing computer assistance, control and high level integrated automation at all levels within manufacturing industries, by linking "islands of automation" into a distributed processing system. Ranky draws a picture of CIM as being composed of four major sub-systems (see figure 2.2), viz: business data processing system, computer aided design, computer aided manufacturing and flexible manufacturing systems (FMS) incorporating machining, assembly, test and other processes. Ranky provides detail activities related to the manufacturing business, including:

* evaluating and developing different product strategies,
* analyzing markets and generating forecasts,
* analyzing product/market characteristics and generating concepts of possible manufacturing systems (i.e. FMS systems),
* designing and analyzing components for machining, inspection, assembly and all other processes relating to the nature of the components and/or product,
* evaluating and/or determining batch sizes, manufacturing capacity, scheduling and control strategies relating to the design and fabrication processes involved in a particular product,
* analyzing feedback of certain selected parameters relating to the
FIGURE 2.2 AN OVERALL CIM MODEL INTEGRATING BUSINESS DATA PROCESSING, CAD, CAM AND FMS (A COPY FROM [RAN86])
manufacturing processes, and evaluating status reports from the DNC (Direct Numerical Control) system,
* analyzing system disturbances and economic factors of the total system.

In summary, certain levels of awareness of CIM exist with regard to what CIM is for and how CIM can be defined. But, the awareness of CIM is far from being widespread and the knowledge of CIM is a long way from being mature.

2.5 Benefits of CIM Systems Integration

2.5.1 How to Justify Investments in CIM

Having realised the challenge and potential benefits of CIM, some companies have already invested heavily and the current predictions of the overall growth of this market over the next five to ten years have been quoted as averaging 12% to 15% [BEN85]. Investment cost/benefit analysts also point out that traditional methods for investment analyses, such as payback period, average rate of return, internal rate of return, and net present value methods, are not entirely appropriate for justifying CIM investment [BEN85] [HAR79].

Works [WOR85] suggests that CIM benefits fall into two categories:
(1) traditional cost benefits (direct, objective, quantitative, financial payoffs to the company),
(2) nontraditional noncost benefits (indirect, subjective, qualitative, and personal to CIM user groups).

Figure 2.3 illustrates specific objective (direct) and subjective (indirect) benefits which can be used to justify CIM in a company [WOR85].

-14-
(1) traditional direct cost benefits,
(2) product quality benefits (and hence less rework),
(3) short project span time (i.e. fewer hours per unit shipped and on-line delivery),
(4) integrated database (facilitating reduced paperwork burden through good communication),
(5) a risk reduction strategy for the future,
(6) undocumented, unexplainable cost reductions (i.e. personal payoffs to users).

Figure 2.3 Direct and Indirect Benefits of CIM

Works [WOR85] believed that if top managers gained some understanding of the benefits, they would agree to fund CIM.

2.5.2 Real Integration Benefits

Although full scale and generic CIM systems are non-existent [SHR85] [KOC86] [KRI87], there are some small scale and specific CIM systems. Good examples are the USA Air Force ICAM (Integrated Computer-aided Manufacturing) project [KOC86], the CIAM system (Computerized Integrated and Automated Manufacturing) for the requirements of the aircraft industry [KOC86] and CIFM-85 (Computer Integrated Flexible Manufacturing, 1985) implemented by Ingersoll Milling Machine Co. in the USA [HES85]. In the CIFM-85 system, a spectrum of production sub-systems were integrated [HES85]. These include:

* Master schedule,
* Engineering design,
* Production planning and control,
Inventory control,
* Purchasing and accounts payable,
* Routing and process planning,
* Numerical control programming and post processing,
* Flexible machining system,
* Parts storage and retrieval, automatic transportation, part identification and tracking, DNC, automatic inspection, tool and fixture management, and process data management and report,
* Assembly,
* Job cost and management reports.

The potential benefits of full scale CIM can be perceived through benefits achieved by small scale CIM installations worldwide. In 1984, the National Research Council (NRC) in the USA published a report on the average benefits achieved through interfacing and integration based on implementation at five major companies (McDonnell Aircraft Co., Deere and Co., Westinghouse Defence and Electronics Centre, General Motors and Ingersoll Milling Machine Co.) [NRC84]. Those benefits are listed below in figure 2.4.

2.6 Modern Integration Tools and Methods

The potential benefits of CIM are evidently encouraging and there have been significant research activities and investment world-wide aimed at turning "CIM ideals" into "CIM reality". However, despite the availability of a few small scale and specific integrated manufacturing systems, how they were achieved has not been reported in detail to the author's knowledge. Such a lack is probably due to keeping high-technology secret for gaining a competitive edge. In view of
the currently continuous evolution of MAP/TP, Protocol), we can say that CIM systems integration is a relatively new research area with as yet little generic or conclusive knowledge available. Part of the reasons can be due to the fact that the scope of the area is enormous. Hence, the process of searching

(1) Reduction in engineering design cost 15-30%
(2) Reduction in overall lead time 30-60%
(3) Increased product quality as measured 
   by yield of acceptable product 2-5 times previous level
(4) Increased capability of engineers as measured 
   by extent and depth of analysis in same or 
   less time than previously achieved 3-35 times
(5) Increased productivity of production operations 
   (complete assemblies) 40-70%
(6) Increased productivity (operating time) of 
   capital equipment 2-3 times
(7) Reduction of work in process 30-60%
(8) Reduction of personnel costs 5-20%

Figure 2.4 Benefits achieved through interfacing and integration efforts

through and classifying relevant research achievements is a contribution in its own right. Modern integration tools and methods to be reviewed will be categorized here under the following three headings:

(1) industrial computer networks,
(2) manufacturing information format standardization,
(3) systems integration architecture.
2.6.1 Industrial Computer Networks

Here the phrase "industrial computer networks" will be used to encompass Local Area Networks (LANs), Wide Area Networks (WANs) and point to point digital data links, among which LANs are arguably of primary importance for CIM systems integration. LANs, which are concerned with the interconnection of distributed communities of computers and computer controlled equipment within a single building or localised group of buildings [HAL85], provide a fundamental communication technology for CIM [RAN85] [GRO87].

A. ISO/OSI MAP/TOP

Communication problems in large manufacturing companies such as General Motors and the Boeing corporation are more complex and difficult to solve than in small companies. GM claimed that in 1985 they were using some 20000 programmable controllers and 2000 robots, and that there will be a 400% to 500% increase in the population of "intelligent" manufacturing devices expected in 1990. Only 15% of the processes using these devices in 1985 had facilities for communication beyond their own islands of automation [PLA87] [WES87]. The Boeing corporation is similarly faced with complex communication problems in their office environment [PLA87] [WES87]. Being fundamental to CIM, communication between computers and computer-controlled devices from different vendors must be accomplished [DTI87] [GRO87].

The problems associated with achieving communication between heterogeneous computers and computer-controlled machines, and the potential benefits of CIM, have stimulated GMs' and the Boeing corporation's commitment to enforcing
their equipment suppliers to provide "standard" communication facilities. As early as in 1979, a MAP task force was created by the Advanced Product Manufacturing and Engineering Staff (APMES) at General Motors' Technical Centre [MAP86] to specify "standard" communication protocols, wherever possible being based on generally accepted communication standards and procedures for shop floor machines (robots, PLCs, NCs, CNCs, etc) and to encourage vendors' acceptance of MAP. Resulting from the work of the MAP task force, GM recognized the need for other companies to support MAP. Therefore a MAP Users Group was formed in 1983 to build up the MAP "strength" [HER86].

MAP for the interconnection of shop floor devices was mirrored by TOP for the interconnection of computers in office environments, which was started by Boeing company [HER86] [WES87]. It was announced, in September 1985, that the MAP and TOP steering committees were to be combined into a joint MAP/TOP steering committee to work towards a common goal [WES87] [STU88].

The framework chosen for MAP/TOP is the International Standards Organisation/ Open System Interconnection (ISO/OSI) reference model [ZIM80] [ETT87] which is an internationally agreed seven layer structure (see figure 2.5). The reference model defines the functionality required at each layer to establish a reliable and timely delivery of messages between two or more computers. MAP/TOP has evolved through several versions [WES87] [ETT87] [MIP87]:

* The first MAP document in 1982,
* MAP version 1.0 in April 1984,
* MAP version 2.0 in February 1985,
<table>
<thead>
<tr>
<th>Layer 7</th>
<th>Layer 6</th>
<th>Layer 5</th>
<th>Layer 4</th>
<th>Layer 3</th>
<th>Layer 2</th>
<th>Layer 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Presentation</td>
<td>Session</td>
<td>Transport</td>
<td>Network</td>
<td>Data Link</td>
<td>Physical</td>
</tr>
</tbody>
</table>

**VIRTUAL CONNECTION**

**COMPUTER STATION X**
- User Application Process
- Layer 7 Application
- Layer 6 Presentation
- Layer 5 Session
- Layer 4 Transport
- Layer 3 Network
- Layer 2 Data Link
- Layer 1 Physical

**COMPUTER STATION Y**
- User Application Process
- Layer 7 Application
- Layer 6 Presentation
- Layer 5 Session
- Layer 4 Transport
- Layer 3 Network
- Layer 2 Data Link
- Layer 1 Physical

**Transmission Medium**
- Co-ax or Fibre Optic

**User Application Software**
- not part of the ISO model

**Provides Network Services**
- to the Application Processes

**Formats the data to/from the Network into a standard format**

**Synchronisation & Data Management, Name/Address, Translation, Access Security**

**Provides Transparent Data Transfer between End Systems**

**Accomplishes Message Routing to/from Stations on other Networks**

**Accomplishes Message Routing, Error Detection, Retries Between Stations on Local Networks**

**Encodes/Decodes Messages and Transfers Data to/from the Transmission Medium**

**Figure 2.5 ISO/OSI Reference Model**
* MAP version 2.1 in March 1985,
* MAP version 2.2 in September 1986,
* MAP version 3.0 in July 1987,
* TOP version 1.0 in November 1985,
* TOP version 3.0 in July 1987 (no TOP version 2.0).

Here the protocols of the latest MAP/TOP version 3.0 are summarised in figure 2.6.

Beale suggests that MAP/TOP has now become the most significant user initiative in the history of computing [BEA88]. Stimulated by GM and Boeing, 21 vendors including IBM, AT&T and Siemens, have invested an estimated $70 million on developing the OSI-based software and hardware to demonstrate MAP/TOP at Autofact'85 (Detroit, USA) in November 1985 [BEA88] [WES87] [STA85]. Large sums of money were also involved in the CIMAP (Computer Integrated Manufacturing MAP) which was held at Birmingham, UK, in December 1986 with some 54 UK companies/organizations demonstrating MAP products in some form [WES87]. The CIMAP event demonstrated that an estimated 60% of the costs for a typical CIMAP demonstration cell were for systems integration software [HOL87] as depicted in figure 2.7.

The US MAP/TOP Users Group has over 250 companies and the European MAP Users Group (EMUG) has about 100 companies, including BP, Daimler-Benz, FIAT, Philips, Renault, Shell and Unilever [BEA88] [WES87]. Significant interest in MAP/TOP has also been shown by companies from Japan and Australia [WES87].

After several years of enhancement and extensions, the MAP/TOP specifications
<table>
<thead>
<tr>
<th>Layer 7</th>
<th>TOP</th>
<th>MAP</th>
<th>Layer 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>CCITT X.400 MHS</td>
<td>ISO DIS 6571 FTAM</td>
<td>IEEE 802.3 IEEE 802.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CSNA/CD TOKEN BUS</td>
</tr>
<tr>
<td>Layer 6</td>
<td>CCITT X.409 SYNTAX</td>
<td>ISO DIS 8823 KERNEL</td>
<td>IEEE 802.5</td>
</tr>
<tr>
<td>Presentation</td>
<td></td>
<td>ISO 8824/5 ASN.1 SYNTAX</td>
<td></td>
</tr>
<tr>
<td>Layer 5</td>
<td>ISO IS 8327 BASIC COMBINED SUBSET</td>
<td>ISO IS 8073 CLASS 4</td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer 4</td>
<td>ISO IS 8473 CLNS INTERNET</td>
<td>ISO IS 8802/2 LLC TYPE 1</td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer 3</td>
<td>ISO IS 8473 CLNS INTERNET</td>
<td>ISO IS 8802/2 LLC TYPE 1</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer 2</td>
<td>ISO IS 8802/2 LLC TYPE 1</td>
<td>ISO IS 8802/2 LLC TYPE 1</td>
<td></td>
</tr>
<tr>
<td>Data Link</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.6 MAP/TOP Version 3.0 Layer Standards
FIGURE 2.7 COST BREAKDOWN FOR A TYPICAL CIMAP CELL
have been developed into the latest version 3.0, in which the application layer (layer 7) supports functions like FTAM (File Transfer, Access and Management), MHS (Message Handling System) and MMS (Manufacturing Message Services). FTAM serves to transfer files in binary or ASCII data formats, to create and delete files, and to read and update contents and characteristics of a file [STU88]. MHS (CCITT X.400) is in fact an electronic mailing system [VER86] [MIP87]. It should be noted, however, that application processes cannot meaningfully exchange information by using FTAM and MHS services [MIP87].

MMS, which resides in the application layer of MAP [EIA87], is a communication protocol standard for manufacturing applications. Although MMS is applicable to a wide range of plant applications, unlike its predecessor (Manufacturing Message Format Standard) the standard does not contain information on specific applications [EIA87] [MIP87]. Actually, MMS is seen as a generic standard in that it specifies a general methodology for how messages are assembled and sent, i.e. it provides structures and coding procedures for manufacturing messages [EIA87] [MIP87]. Its companion standards will specify additional specific message standards for industrial shop floor machines (e.g. robots, PLCs, NCs) [EIA87].

B. SNA, Ethernet, X.25, etc

In spite of the strength of the MAP/TOP initiative, it is unlikely that all existing proprietary LANs will become obsolete. For example, in the foreseeable future it is extremely unlikely that IBM’s Systems Network Architecture (SNA) will be replaced by OSI implementations [MAI88] because of its maturity in
architecture, its functional capabilities, (*) and available support products. Over 20,000 SNA mainframes throughout the world have been installed involving vast investment in SNA-based application software [MAI88]. Thus, it must be accepted that SNA, along with other network protocols such as Ethernet, X.25, and Cambridge ring, will coexist with MAP/TOP in enterprises. The importance of other network architectures necessitates some knowledge of the issues involved as summarised below.

SNA

IBM's Systems Network Architecture (SNA) was announced in 1974, being also based on a seven layered architecture designed to provide interconnection between IBM products [SCH87]. Since SNA preceded ISO/OSI developments, SNA architecture does not conform, layer by layer, to ISO/OSI reference architecture. Figure 2.8, however, illustrates an approximate correspondence between SNA layers and those of the ISO/OSI reference model [TAN81].

Ethernet

Ethernet version 1.0 was released jointly by the Digital, Intel and Xerox corporations in May 1980. Ethernet services correspond to the physical and data link layers of the ISO/OSI reference model [CHE83] [ETH80] [MIE84]. In short, Ethernet is a 10 Mbits/s baseband coaxial cable-based local network that uses (*) The functional services offered in current SNA installations exceed those of TOP, but this is unlikely to remain as a steady state in the future when TOP becomes very widely accepted and utilised.
FIGURE 2.8 APPROXIMATE CORRESPONDENCES BETWEEN SNA AND ISO/OSI REFERENCE MODEL

<table>
<thead>
<tr>
<th>ISO/OSI</th>
<th>SNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAYER 7</td>
<td>END USERS</td>
</tr>
<tr>
<td>APPLICATION</td>
<td></td>
</tr>
<tr>
<td>LAYER 6</td>
<td>NAU SERVICES</td>
</tr>
<tr>
<td>PRESENTATION</td>
<td></td>
</tr>
<tr>
<td>LAYER 5</td>
<td>DATA FLOW CONTROL</td>
</tr>
<tr>
<td>SESSION</td>
<td></td>
</tr>
<tr>
<td>LAYER 4</td>
<td>TRANSMISSION CONTROL</td>
</tr>
<tr>
<td>TRANSPORT</td>
<td></td>
</tr>
<tr>
<td>LAYER 3</td>
<td>PATH CONTROL</td>
</tr>
<tr>
<td>NETWORK</td>
<td></td>
</tr>
<tr>
<td>LAYER 2</td>
<td>DATA LINK CONTROL</td>
</tr>
<tr>
<td>DATA LINK</td>
<td></td>
</tr>
<tr>
<td>LAYER 1</td>
<td>PHYSICAL</td>
</tr>
<tr>
<td>PHYSICAL</td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 2.9 TYPICAL TRANSMITTED ETHERNET FRAME
an access-control method called Carrier-Sense Multiple-Access with Collision Detection (CSMA/CD). Ethernet has been accepted as an IEEE communication standard, namely IEEE 802.3 [MIE84]. A typical Ethernet frame [CHE83] [MIE84] transmitted across an Ethernet LAN is shown in figure 2.9. The physical and data link services of TOP are based on Ethernet protocols.

X.25 [SCH87]

X.25 was approved by the CCITT as a network interface recommendation in 1976 and was revised twice in 1980 and 1984 respectively. X.25 can be employed to connect data terminals, computers, and other user systems or devices to packet-switched networks (or WANs) and is organized as a three-layered architecture with services provided corresponding to the bottom three layers of the ISO/OSI model as depicted in figure 2.10.

Kermit [CRU84]

A proprietary communication protocol package, known as Kermit, will be considered here as it was used in this research study. Kermit was originated in early 1985 at Columbia University in the USA to satisfy economically the need for file transfers between two computers. The impetus for its development was that commercial LAN products were expensive and at that time not yet widely available. Kermit is based on the use of an RS 232c serial data link. With reference to the ISO/OSI reference model, Kermit, although simple, can be viewed as comprising four layers, namely: the physical, the data link, the session and the application layer. Whereas the physical layer is implemented based on the use of RS 232c protocols, the mechanisms involved in the other three layers
are realized by a Kermit packet of information as shown in figure 2.11.

C. Interworking of Various Networks

Having realized that other network protocols would continue to be used, the MAP/TOP initiative advocates the use of four types of communication relay called repeaters, bridges, routers and gateways. These four relays can be used to interconnect networks of the same or different types including MAP with TOP [BOE85] [MAP86]. The functions of these four relays are listed in figure 2.12 and their architectures are depicted in figure 2.13.

2.6.2 Manufacturing Information Format Standardization

In order to reflect the author's view concerning the definitions of "data" versus "information", the subtitle "manufacturing information format standardization" is used instead of the commonly used term "manufacturing data format standardization". In the author's opinion, "data" are symbols which have no meaning or whose meaning is of no concern to users. However, "information" can be defined as symbols which are meaningful to users for describing the real world. "Information" is inseparable from its use. Hence, the common terms "CAD/CAM data" ought to be called "CAD/CAM information" which refers to the meaningful symbols relating to CAD/CAM. Conventionally, the term "information" is rarely used while the term "data" is frequently used even if values of "symbols" are of concern. The author suggests that the term "information" should be used when values of "symbols" are of concern because this change will enhance readers' interests in information technology. Wherever appropriate in the thesis, "information" will be used to replace "data". But, two
Figure 2.10 Relation Between ISO/OSI and Architecture

Figure 2.11 The Format for A Packet of Information According to The Kermit Protocol
<table>
<thead>
<tr>
<th>Relay type</th>
<th>Transparency</th>
<th>Type of networks to be connected</th>
<th>At which level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeater</td>
<td>Yes</td>
<td>One type of network segment</td>
<td>Physical</td>
</tr>
<tr>
<td>Bridge</td>
<td>Yes</td>
<td>Networks which use consistent addressing scheme and frame size at data link level</td>
<td>Data Link</td>
</tr>
<tr>
<td>Router</td>
<td>No</td>
<td>Different types of networks up to network level</td>
<td>Network</td>
</tr>
<tr>
<td>Gateway</td>
<td>No</td>
<td>Different types of networks up to application level (e.g., OSI with non-OSI)</td>
<td>Application</td>
</tr>
</tbody>
</table>

**FIGURE 2.12 RELAY TYPES AND THEIR FUNCTIONS**
FIGURE 2.13 COMMUNICATION RELAYS: A, REPEATER; B, BRIDGE, C, ROUTER, D, GATEWAY
very popular terminologies "database" and "data model" will be kept although the author is in favour of the terminologies "information base" and "information model".

Among the seven layer protocols of MAP/TOP, presentation layer protocol was not chosen or defined prior to MAP/TOP version 3.0 released in 1987. The presentation layer services of MAP/TOP version 3.0 deal only with the transformation of basic data types (e.g. integer, floating point reals) [MIP87]. From an ISO/OSI's perspective, all information translation should be included as presentation layer activities [ZIM80]. However, there are various current information format standardization activities which in general are aimed at solving the common problem through viewing the requirements in a different way. The following sub-sections consider such activities.

A. The Reason for Manufacturing Information Format Standards

A variety of information format standard initiatives have been invoked, which essentially have a similar overall aim of economically facilitating the information transfers between heterogeneous computer systems. For example, there are many proprietary CAD and production management packages, each having their own information formats [HIL86]. However, in almost every user company, a multi-vendor environment exists which makes it impossible to electronically transfer information between different software systems, especially between CAD systems, when enormous diversity in information formats exists [CHO88]. To solve information interpretation problems, two approaches can be taken, namely: (1) a "specialized information translator" approach and (2) an "information standard" approach (see fig. 2.14). The philosophy underlying
FIGURE 2.14 TWO APPROACHES FOR INFORMATION TRANSLATION

A. Specialized information translator approach

B. Information standard approach
various information format standard initiatives is the same and can be illustrated by the information transfer requirements between n dissimilar CAD systems. Whenever it is necessary to transfer information between two dissimilar CAD systems, a translator must be written. If specialised translators are written to allow information to be transferred between n CAD systems, a total of n(n-1) translators will be needed \cite{CHO88} \cite{PRA85}. The number of translators required will increase in proportion to the square of n. However, if a CAD information format standard is used, the number of translators for information transfer between n CAD systems will be 2n. (i.e. the number increases linearly with n \cite{CHO88} \cite{PRA85}). It can be seen that an "information standard" approach is a more practical and economical way to solve the information interpretation problem, especially where an increased level of complexity is anticipated and where the need to reconfigure a CIM system is expected over its lifetime.

B. Major National/International Manufacturing Information Format Standards

IGES

A number of national and international standards organizations have been developing information format standards for CAD/CAM. An early major CAD/CAM information format standardization movement was started in the late 1970's by the US Department of Defense to develop IGES (Initial Graphics Exchange Specification) \cite{CHO88} \cite{PRA85}. This initiative arose from the need to transfer design information between different parts of large companies or from companies' headquarters to subcontractors. The name IGES can be misleading as IGES was developed as a neutral format for the transmission between dissimilar
CAD/CAM systems, of engineering drawings coded electronically, which in fact contain not only graphical information but also textual information [PRA85].

An IGES file is written in ASCII characters and comprises five sections, viz: (1) start section, (2) global section, (3) directory entry section, (4) parameters data section, and (5) terminate section. A detailed description of each section can be found in the literature [CHO88].

SET

While IGES is a de facto standard and is most widely used, especially in USA, the second most widely implemented standard is Standard d’Echange et de Transfer (SET) which covers the about same range of entities as IGES and is under consideration as a French national standard [OWE87] [SET84].

XBF

XBF (eXperimental Boundary File) was developed by the international organization CAM-I (Computer Aided Manufacturing - International, Inc) as a means of transferring solid modelling information [OWE87].

PDDI

The Product Data Definition Interface (PDDI) was developed under the US Air Force ICAM (Integrated Computer Aided Manufacturing) program and can be used to transmit design information as well as manufacturing-oriented information [OWE87].
PDES/STEP

Since 1985, experiences gained in the development of IGES, XBF, PDDI, SET and various other standards have been used by the IGES community in the USA to develop a more comprehensive information exchange standard named Product Data Exchange Specification (PDES). PDES contains a very wide spectrum of product modelling information, which includes not only product definition information but also product life cycle information (e.g. concerning analysis, manufacturing, quality assurance, testing and product support), application-specific administration and control information [OWE87] [PDE86]. The development of PDES has been mirrored by the development of STEP (STandard for the Exchange of Product model data) of the ESPRIT CAD*1 project in Europe [OWE87]. Both PDES and STEP are similar in architecture: PDES utilising a three schema architecture [PDE86] and STEP with a layered architecture [OWE87] comprising (1) application layer, (2) logical layer, and (3) physical layer. The PDES and STEP communities were reported to be committed towards a joint standardization programme called PDES/STEP [OWE87].

EDIF

Standardization processes are usually very time-consuming partially because different standardization communities often proceed on their own. For example, the separation of EDIF (Electronic Design Interchange Format) from the IGES/PDES/STEP initiative is largely for political reasons [OWE87]. EDIF was originally developed by six large US electronics companies and with the support
of European representatives it is rapidly becoming a de facto standard for the exchange of electronic CAD design information [HIL86]. EDIF files are virtually sequential files of ASCII characters based upon the LISP programming language. An incomplete example EDIF file is listed as the following [HIL86]:

(EDIF example file

(status

(EDIF version 1 1 0)
(EDIF Level 0)

(written

(timestamp 1986 2 4 11 11 39)
(accounting Author "Keith Benett")
)
(library example library

(comment "a library contains the definition of any number of cells")
)

(comment "case is only significant in strings")
(comment "end of the EDIF file")
)

DMIS

One more information standard worth mentioning is the DMIS (Dimensional Measuring Interface Specification) which allows inspection information to be transferred between CAD systems and inspection equipment. DMIS was created under the auspices of Computer Aided Manufacturing International (CAM-I) in the USA through the Quality Assurance Program and is now being promoted as
Despite the availability of various widely implemented information standards, no single international manufacturing information standard exists. However, the PDES/STEP initiative may resolve this situation to a certain extent in the future [STE87]. Stewart [STE87] emphasizes the importance and urgency of information standards in the integrated CAD/CAM environment. Stewart [STE87] also points out that the development of standards would not be possible unless CAD/CAM users groups, CAD/CAM vendors and national/international standardization organizations are willing to cooperate with each other. Furthermore, the information standards in other areas such as those for production planning and control modules are less well developed, when compared with design information standards.

2.6.3 Systems Integration Architecture

As previously stated, implementation experience at the CIMAP event has shown that the systems integration software costs commonly account for 60% of the total cost of an integration demonstration cell based on MAP [HOL87]. There are huge gaps existing between current MAP/TOP specifications and fully "open" CIM systems [STU88]. For evolving generic integration tools, the "three architecture" approach to systems integration was proposed by Weston, et al. [WES87] [GAS87] [SUM87] as illustrated in figure 2.15. The architectures were (1) network architecture, (2) information architecture, and (3) application architecture respectively.

The functions performed by those architectures and their relationships were
FIGURE 2.15 SYSTEMS INTEGRATION ARCHITECTURE
described either informally or implicitly. Gascoigne et. al. [GAS87] commented that the MAP version 2.1 was aimed primarily at defining the "network architecture" whereas associated "application" and "information" architecture were not yet defined. Weston et. al.[WES87] widened the scope of the "network architecture" to encompass TOP as well as MAP and also stressed that the "network architecture" must be complemented by an "application architecture" and a "database architecture" (used interchangeably with information architecture). The case of the Automated Manufacturing Research Facility (AMRF) in the USA NBS (National Bureau of Standards) was considered as an example of studying a variety of application architectures which implied the use of the hierarchical control models proposed by AMRF [BOLS83] (see fig. 2.16) within the "application architecture". The role of the information architecture was informally defined through reference to the Integrated Information Support System (IISS) created at the USA Air Force Wright Aeronautical Laboratories [ICA83].

Sumpter, et.al. [SUM87] presented a more detailed definition of the three architectures: "network architecture" is to support information exchange, "application architecture" is to implement system supervisory and management functions, and "information architecture" is to provide database and offline design facilities. Further the application architecture was described by Sumpter, et. al. [SUM87] as being a structured environment within which application processes were executed. The relationship between network architecture and application architecture was stated quite clearly as the former provided the services of information/data exchange between "application processes" [SUM87]. However, questions remain concerning how the "application architecture" interacts with the "information architecture" and their detailed functionalities.
FIGURE 2.16 AMRF CONTROL HIERARCHY
FIGURE 2.16 AMRF CONTROL HIERARCHY
Nevertheless, "three architecture" approach represented a significant step forward towards achieving "open" CIM systems integration because it decomposes the complex integration problem into more manageable and logically separate sub-systems.

2.7 Summary

There is a growing awareness of the potential benefits of CIM systems integration, along with the encouragement from the real benefits achieved in some of the small scale and specific integrated manufacturing systems world-wide. MAP/TOP initiatives have injected momentum into the CIM systems integration movement. Although the communication via industrial computer networks, arguably mainly MAP/TOP in the future, has been recognized as fundamental to CIM systems integration, the functions of MAP/TOP and other industrial computer networks required for full scale CIM systems are not yet clear. There is a lack of independent assessment of MAP/TOP, which, in the author's opinion, is important both for MAP/TOP users and for future MAP/TOP developments. Chaos exists concerning manufacturing information format standardization, which necessitates the evaluation of the role of various information format standardization initiatives. Information is widely recognised as being important, but how important information is in the overall scenario of CIM systems integration and how information automation can be achieved in a generic way remain unanswered questions. The CIM systems integration architecture is of paramount importance because it is the framework for systems integration projects. Previous work has shown the need to decompose the integration problem into manageable sub-problems (i.e. "three architecture" approach), and to utilize emerging standards as appropriate. The proposal of
"three architecture" approach was indeed a step forward towards the design of the CIM systems integration architecture. However, as reviewed previously, prior to this research study it lacked substantiation and clear specification. In other words, the formalisation of such an important integration architecture requires more support, substantiation and more precise specification. In short, despite the extremely high level of investment made in this area, CIM systems integration is a relatively new research area with little generic and conclusive knowledge available.
CHAPTER 3
AN EARLY STAGE INTEGRATION ACTIVITY AND PROJECT SPECIFICATION

3.1 Introduction

Many major problems remain in the area of CIM systems integration as illustrated by the literature survey. Certain of those problems can be highlighted by asking the following three questions:

(1) what are the necessary properties of a formal architecture for systems integration?

(2) what potential roles can computer network (e.g. MAP/TOP) and manufacturing information format standards play?

(3) what kind of integration tools and methods are required?

It is extremely difficult to answer each of these important questions and to provide solutions, which can provide a migration path to full CIM systems integration.

This chapter will describe the specific integration tool produced, in the early stages of this work, for integrating a PP&C (Production Planning and Control) software package with a FAC (Flexible Assembly Cell) controller. This achievement widened the author's vision about CIM systems integration, enabling generic project objectives to be determined.

3.2 Integrating a PP&C Package with a FAC Controller
3.2.1 Introduction

The integration of a PP&C software package with a FAC controller was chosen as an early stage integration activity. A primary aim of this study was to gain in-depth understanding of the integration problems raised by questions (1) to (3) above. The approach taken, in this initial study period, is a logical one and corresponds to the most likely approach that would be taken by systems integrators in industry. It is natural to begin by considering how to integrate the activities of only two manufacturing sub-systems, such as, a PP&C software package with a FAC controller, CAD and CAPP, CAD and ATE (Automatic Testing Equipment), CAPP and MRP, MRP and scheduling.

3.2.2 Facilities Utilised

A SUN workstation (with UNIX multi-tasking operating system) and an IBM PC AT (with single-tasking operating system) were the computers used in this initial integration activity. The KERMIT file transfer facility [CRU85a] [CRU85b] was used to establish the data link between the two computers since MAP/TOP communication facilities were not available at that time. The PP&C software package used was the MICROSS package [KEW85] which was installed on the IBM PC AT. A FAC controller was being developed in-house at the SUN workstation by other researchers in the L.U.T. (Loughborough University of Technology) Systems Integration Group. All facilities are depicted in figure 3.1.

With reference to the literature survey, FTAM of MAP/TOP version 3.0 can be used to accomplish data file transfers. The KERMIT file transfer facility (see Appendix A) provides similar functionality but only between two computers and
FIGURE 3.1 FACILITIES FOR INTEGRATING PP&C SOFTWARE PACKAGE WITH A FAC CONTROLLER
at a lower data rate. KERMIT is widely available and can run under the operating systems of the computer hardware used in this study. At a later date, KERMIT could have been replaced by MAP/ TOP FTAM. The elaboration of this point can be found in chapter 9.

The MICROSS PP&C software package purchased for this project is typical of commercial PP&C packages with regard to the man-machine interface facility it offers. Designed in modular form, the version 3.27 MICROSS package [KEW85] contains software modules for production control, material requirements planning, stock control and bill of material. It is menu-driven, being arranged in a hierarchy as shown in figure 3.2.

In order to integrate the MICROSS PP&C software package with a FAC controller, the man-machine interface of MICROSS needs to be studied. Here, we will consider an example of obtaining work-to-lists to illustrate how an operator can communicate with MICROSS, the process of which is depicted in figure 3.3 where work-to-lists are displayed on the screen. If the operator mistypes any number (for example, work centre code "101" as "M01"), MICROSS has the built-in capability to validate the input information items and to reject invalid ones. When invalid information items are rejected, the package will wait for the correct ones or expect to be returned to the main menu and subsequently the operating system. The information integrity is thus maintained. In brief, an operator can easily retrieve and update information stored in MICROSS and execute decision-making programs (e.g. scheduling and MRP) by selecting numbers for the hierarchical menus and responding to a list of questions/answers, with the whole man-machine interface process being interactive.
FIGURE 3.2 EXAMPLE HIERARCHICAL PP&C MENUS
FIGURE 3.3 AN EXAMPLE OF AN OPERATOR'S COMMUNICATION WITH MICROSS

1. PC MENU
2. JOHN
3. 310
4. D
5. 101
6. 101

OPERATOR

MICROSS

ENTER PASSWORD

PC MAIN MENU
100 ...
120 ...
...
310 WORK-TO-LISTS
...

OPTION:

DISPLAY, PRINT
OR MAIN MENU
(D,P or H)

FROM WORK CENTRE NO.

TO WORK CENTRE NO.

WORK-TO-LISTS FOR WORK CENTRE NO. 101

- 49 -
3.2.3 Automating Remote Information Access

Many types of information are stored in MICROSS, work-to-lists being a typical example. A FAC controller depends on work-to-lists to coordinate and control the activities of machines in the FAC. It also generates work-in-progress information by monitoring and processing the information created at machines in the FAC, and the information is to be stored in the PP&C package. The integration of a PP&C software package with a FAC controller can be interpreted generally as automating information updates and retrievals between the PP&C package and the FAC controller. The specific issues are illustrated in the following.

With the facilities available in proprietary form, it was not possible for the FAC controller to electronically access information stored in the MICROSS package. A desirable feature would be to provide the FAC controller with meaningful PP&C commands as the FAC controller had no access to the menus of MICROSS. Meaningful PP&C commands would allow a programmer to efficiently define information access at the FAC controller. However, the specific proprietary MICROSS software package could only understand commands in numbers, such as, "310" for obtaining work-to-lists. Therefore, a "command translator" was required to carry out command translations, which directly communicated with MICROSS in numbers and should be installed on the IBM PC AT. A consistent set of PP&C commands could be embedded in the in-house developed FAC controller. It was also necessary to have a "command communication agent" installed on each computer. The agent at the SUN workstation accepted commands issued by the FAC controller, called KERMIT to pass the commands to the other agent at the IBM PC AT, received resulting
information via KERMIT and delivered the received information to the FAC controller. The other agent at the IBM PC AT passed the commands originally issued by the FAC controller to the command translator, collected the resulting information from the MICROSS package and delivered the resulting information to the command communication agent at the SUN workstation via KERMIT. Once the FAC controller obtained the information, it could use the information either (1) to make decisions, or (2) to coordinate the activities of machines in the FAC. This specific integration solution for automating remote information access between the MICROSS package and a FAC controller is depicted in figure 3.4.

The PP&C commands which have been implemented are:

1. **LISTMACH** (list all machines),
2. **GETWKLT** *wk-code* (get work-to-lists for a machine),
3. **WIPDOP** *job-no net-ref op-no* (report an operation completion for a particular job and component, which is one example of work-in-progress updates),
4. **LISTJOB** (get a list of all jobs),
5. **LISTCMP** *job-no* (get a list of components for a particular job),
6. **LISTOP** *job-no net-ref* (get a list of an operation detail for a particular component of a particular job),
7. **JOBSTATUS** *job-no* (get job status for a job).

If both a cell controller and a PP&C software package are designed by different vendors, it is extremely unlikely that they can generate and understand information in the same format. As a result, an information translator is also needed, which can be installed either on the SUN workstation or on the IBM PC.
FIGURE 3.4 SPECIFIC INTEGRATION TOOL
3.3 Limitations of Pair Integration

In this thesis, the integration of any two separate manufacturing sub-systems is referred to as "pair integration". The above integration of a PP&C software package with a FAC controller is one specific example of such a pair integration. Pair integration can be accomplished in a similar way as this integration activity for any two interrelated manufacturing sub-systems, such as CAD and CAPP, CAD and ATE, CAPP and MRP, MRP and scheduling. The main components of the pair integration tool are "command communication agent", "command translator" and "information translator", which are not conceptually complicated but can be very time-consuming to produce.

If an enterprise is electronically integrated using this pair integration approach, there are many limitations as the following.

(1) It is inefficient and inconvenient. This is because
   (a) if one manufacturing sub-system has to access information stored in n other manufacturing sub-systems, it must use n different pair integration tools to access the information fragments,
   (b) if one manufacturing sub-system has information needed by n other sub-systems, and information accesses can only take place in sequence through manually activating n different pair integration tools, even when using a multi-tasking operating system.

(2) It is unreliable because there is no inherent facility to enable recovery from system failures when commands are accepted but not yet processed.

(3) It is costly because repetitions of functionally similar pair integration tools
cannot be avoided.

(4) It cannot be easily expanded. This is because

(a) it cannot accommodate changes of manufacturing sub-systems because newly introduced manufacturing sub-systems, which are to replace old ones, require new specific pair integration tools to be produced;

(b) information services are neither provided on a network-wide basis for all managers and operators nor for manufacturing sub-systems which will be installed on computers which have no specific pair integration facilities.

Therefore, we can say that although the achievable specific pair integration approach is useful for realising systems integration, it has limitations with regard to the levels of integration achieved and the costs involved. None-the-less, in the short term, systems integrators will continue to integrate their systems in this way as it is perceived as being expedient. However, this approach can only be used sensibly for simple systems and its proliferated use will severely limit the evolution of flexible and full scale CIM systems, and thus affect the future profitability of manufacturing enterprises.

3.4 Project Specification

The success of the early stage integration activity served to significantly widen the author's vision concerning systems integration problems, and led to the specification of generic project objectives, as described in the following:

(1) to formalise the design of a flexible and generic architecture for systems integration,
(2) to assess the roles of MAP/TOP standards and manufacturing information format standards,

(3) to develop generic integration tools and methods.

The formalisation of the design of an overall systems integration architecture (or infrastructure) is of paramount importance with regard to long term implications for achieving CIM. MAP/TOP networks and manufacturing information format standards represent evolving and currently available integration tools and methods in some form which have a measure of acceptance in the world-wide CIM area. Therefore, the assessment of MAP/TOP standards and information format standards is important both for users of those standards and for possibly influencing the future development of those standards. Generic integration tools and methods are what world-wide systems integrators are searching for, certainly from the viewpoint of manufacturing users. However, it is not possible to solve the whole systems integration problem in a single step. Therefore, the sub-problems will be identified having considered the overall systems integration architecture. Work in achieving project objective (3) is further detailed wherever appropriate in the body of the thesis.

3.5 Summary

The successful integration of a PP&C package and a FAC controller served to greatly widen the author's vision concerning systems integration problems, thus enabling generic project objectives to be determined. The next chapter is to analyse currently stand-alone MEs to be integrated and their interactions in manually integrated systems for formalising the design of flexible and generic systems integration architecture as stated in project objective (1).
CHAPTER 4

ANALYSES OF MANUFACTURING ENTITIES
AND THEIR INTERACTIONS FOR INTEGRATION

4.1 Introduction

CIM systems integration has drawn world-wide research efforts and investment as reviewed in Chapter 2. The commonly used terms like "islands of automation" and "automating information flows" reflect the seriousness of electronic isolation of manufacturing entities and the demands for the automation of information transmission respectively [MIP88].

The significance of CIM dictates the importance of "systems integration architecture" which is the framework for the whole CIM systems integration project. To design this framework was the first project objective determined in the last chapter. Although systems integration architecture was presented as "three architectures" [WES87] [SUM87] as shown in figure 2.15, the proposal was by no means fully substantiated and the precise functional definitions and the relationships between the three architectures did not exist. It must be stated that in order to pursue a generic integration approach rather than a specific pair integration approach presented in the last chapter, it is necessary to study the composition of contemporary or stand-alone manufacturing entities to be integrated, along with their interactions. Here, the composition of a commercial
PP&C package will be analysed first and then this analysis will be extrapolated to reflect the composition of other manufacturing entities. Functional requirements for systems integration tools will be identified by analysing the interactions between MEs in manually integrated systems. This chapter will be concluded with the author's definition of CIM.

4.2 Composition of Manufacturing Entities

By analysing the main menu of the specific MICROSS production control system, it can be found that it possesses two classes of functions as shown in figure 4.1: (1) information retrieval and update (insertion, modification and deletion) function, and (2) decision-making function. The former reflects the property of an information repository, which is for information storage and management. In this case it is contained within the MICROSS production control system in a closely coupled manner. The latter is illustrative of scheduling decision-making capability. Nevertheless, these two parts are not separated because once information has been typed in, scheduling decision-making process can make use of the information "internally". From the viewpoint of information processing, the scheduling decision-making function, whether based on conventional algorithms or artificial intelligence, can be considered as information processing. In this example, the scheduling decision-making function takes information such as a list of machines, jobs, part routings and work-in-progress information and generates information such as work-to-lists. Therefore, the MICROSS production control system can be considered as being composed of an information processing sub-system (scheduling) and a closely coupled information repository.
<table>
<thead>
<tr>
<th>INFORMATION RETRIEVAL &amp; UPDATE FUNCTIONS</th>
<th>DECISION MAKING FUNCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAINTENANCE PROGRAME</td>
<td>SCHEDULING PROGRAME</td>
</tr>
<tr>
<td>100 = SYSTEM DATA &amp; STANDARD CALENDAR</td>
<td>200 = TIME ANALYSIS</td>
</tr>
<tr>
<td>120 = WORK CENTRES &amp; MODIFICATIONS</td>
<td>205 = PRINT JOB PRIORITIES</td>
</tr>
<tr>
<td>140 = WORK IN PROGRESS - INSERT</td>
<td>210 = SCHEDULING</td>
</tr>
<tr>
<td>145 = WORK IN PROGRESS - AMEND/DELETE</td>
<td>220 = WORK TO LIST UPDATE</td>
</tr>
<tr>
<td>150 = WORK IN PROGRESS - REPORT/ENQUIRIES</td>
<td>230 = WORK LOAD UPDATE</td>
</tr>
<tr>
<td>160 = LIBRARY PROGRAMS</td>
<td>240 = RESIDUAL SCHEDULING</td>
</tr>
<tr>
<td>170 = DATA FILES REORGANISATION</td>
<td>250 = AUTOMATIC SCHEDULING</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REPORTS &amp; ENQUIRIES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>300 = WORK LOAD REPORTS</td>
<td></td>
</tr>
<tr>
<td>310 = WORK CENTRE - WORK TO LISTS</td>
<td></td>
</tr>
<tr>
<td>320 = JOB STATUS REPORT</td>
<td></td>
</tr>
<tr>
<td>330 = OVERDUE JOBS REPORT</td>
<td></td>
</tr>
<tr>
<td>340 = DELIVERY PROFILE REPORT</td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 4.1 FUNCTIONS OF MICROSS PC SYSTEM
Similarly, if we examine the MICROSS MRP system, we can find that its functions can also be categorized into the same two classes depicted by figure 4.2. Thus, the statement describing the composition of the MICROSS production control system is also a suitable one for describing the MICROSS MRP system.

In general, a decision-making system requires information and stores information in its information repository, and makes decisions, based on the information available, which are represented also by information and stored in its information repository as well. Decision-making can be viewed as information processing. Therefore, from the viewpoint of information processing, a decision-making system can be considered as being composed of one or more information processing sub-systems and a closely coupled information repository.

Having considered decision-making systems, let us consider the general feature of a CAD system to determine its composition. A product design process can be considered to be a composite decision-making process, involving both a designer and a CAD system and the interactions between them. Product design information generated comprises mainly product geometric information describing the product and is stored in an information repository. The current trend towards including other information types to create a product model can also be viewed in a similar way. As such, a CAD system can also be considered as being composed of one or more information processing sub-systems and a closely coupled information repository.

On a shop floor, a contemporary machine, such as a robot or an NC machine, comprises an electro-mechanical manipulation system and at least one computer which acts as the machine controller. Machines can be used to change the states
<table>
<thead>
<tr>
<th>INFORMATION RETRIEVAL &amp; UPDATE FUNCTIONS</th>
<th>DECISION MAKING FUNCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENQUIRY PROGRAMS</td>
<td>MAINTENANCE PROGRAMS</td>
</tr>
<tr>
<td>200 = STOCK PROFILE ENQUIRY</td>
<td>100 = MATERIAL REQUIREMENTS PLAN - NET</td>
</tr>
<tr>
<td>210 = PLAN PARAMETER ENQUIRY</td>
<td>110 = MATERIAL REQUIREMENTS PLAN - CROSS</td>
</tr>
<tr>
<td>220 = STOCK ENQUIRY</td>
<td>130 = MAINTAIN MASTER PRODUCTION SCHEDULE</td>
</tr>
<tr>
<td>230 = PARTS LIST ENQUIRY</td>
<td></td>
</tr>
<tr>
<td>240 = WHERE-USED ENQUIRY</td>
<td></td>
</tr>
<tr>
<td>REPORT PROGRAMS</td>
<td></td>
</tr>
<tr>
<td>300 = SUGGESTED ORDER REPORT - LIST</td>
<td></td>
</tr>
<tr>
<td>310 = SUGGESTED ORDER REPORT - TABLE</td>
<td></td>
</tr>
<tr>
<td>320 = STOCK REQUIREMENTS PROFILE</td>
<td></td>
</tr>
<tr>
<td>330 = MASTER PRODUCTION SCHEDULE</td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 4.2 FUNCTIONS OF MICROSS MRP SYSTEM.
of the real world. For instance, a component can be moved by a robot from one point in space to another and a hole drilled in a component by an NC drilling machine. Those state changes in the real world result from the execution of machine programs within the machine controller. The machine programs determine the functions of shop floor machines. Often with contemporary machines, the information used in making control and monitoring decisions is tightly coupled to the ME and is commonly stored in read/write memory. However, if MEs of this type are to interact with other MEs in a flexible manner (i.e. are to be flexibly integrated), appropriate access to their information repositories must be enabled. Thus, for systems integration, a manufacturing machine can be viewed as being composed of an information processing sub-system, which makes decisions to enable real world state changes, and an information repository (which with contemporary systems is closely coupled to the information processing sub-system).

Automatic Testing Equipment (ATE) on a shop floor can be viewed as a special class of "information processing and real world state changing sub-system" with a closely coupled information repository. It usually deals with inspecting components, but it may also change real world states such as test voltages and currents, and involve decision-making as a result of processing information. Thus, from the viewpoint of integration, the statement describing the composition of manufacturing machines is also suitable for ATE systems.

The foregoing has conceptually analysed the composition of various contemporary MEs which exist at various hierarchical levels of an enterprise and relate to both design and manufacturing cycles. In summary, for CIM systems integration it is concluded that a contemporary or stand-alone ME can be
perceived as being composed of one or more information processing sub-systems (which makes decisions and/or deals with real world state changes) and an information repository which with contemporary MEs is closely coupled to the information processing sub-system(s). Here the generic term "application program" can be used to encompass both "the pure decision-making sub-system" and "the decision-making sub-system for enabling real world state changes". Information exchange is a means by which all application programs interact with each other. An enterprise can be perceived as being composed of application programs and information repositories as shown in figure 4.3 when considering systems integration. In fact, production processes in an enterprise can be viewed as consisting of discrete and concurrent changes of digital information and the real world states. The problem of integrating stand-alone MEs is really a problem of integrating activities of a variety of application programs and information repositories. The interactions between MEs in manually integrated systems will be analysed in the following section to specify the requirements for systems integration.

4.3 Interactions Between Manufacturing Entities

Contemporary or stand-alone MEs are usually manually integrated, but they are electronically isolated. Thus, a study of the role of operators who accomplish manual integration for realising the interactions between MEs can determine the functional requirements of automatic systems integration tools.

Let us examine a typical ME collection of design, management and control entities. There are logical precedence restraints determining the execution of design, management decision-making and shop floor operations. Product design
FIGURE 4.3 AN ABSTRACT MODEL OF AN ENTERPRISE
is usually done first, then customer orders can be entered. Following the operation of a CAD system, process planning can begin. The start of an MRP system is possible after a master production scheduling system and process planning system have completed their operations and a Bill of Material (BOM) information is available. Subsequently production scheduling processes can begin, following the availability of necessary information required. Nevertheless, those decision-making processes in an office environment are usually not time critical, and the execution precedence constraints of those processes are often not strict and could be concurrent. On a shop floor, examples can be easily found to illustrate precedence constraints between operations. In printed circuit board assembly, hole drilling must be completed before electronic component insertion. Furthermore, insertion cannot be started until the right board has been transported to the right insertion machine. The whole process is illustrated in figure 4.4. Usually precedence constraints on a shop floor are more strict and time critical than in an office environment.

In an office environment, a scheduling operator’s operating procedures can be used to exemplify the roles which operators must play in manual integration. The operator could typically:

1. collect bill of material information from a CAD system,
2. collect process planning information from a CAPP system,
3. collect work-in-progress information from a shop floor information collection system,
4. collect work orders from a MRP system,
5. update a scheduling information repository with information collected above,
6. run scheduling programs to generate work-to-lists,
7. inform shop managers and cell controllers that work-to-lists are ready.
FIGURE 4.4 EXAMPLE PRECEDENCE OF PCB ASSEMBLY EVENTS
Step (1) (2) could be taken very infrequently.

On a shop floor, an example can also be found to illustrate the operators’ typical role. Assuming that three computer-controlled machines, e.g. a hole-drilling machine, a conveyor and an insertion machine, are operated by three manual operators to perform a PCB assembly. Suppose that a PCB is originally placed on the hole-drilling machine and the conveyor is responsible both for component load/unload for machines and for transportation of components. Those three operators can cooperate as follows.

(1) The operator for the hole-drilling machine starts executing a hole-drilling program. When the hole-drilling program is complete, the operator can inform the operator of the conveyor that the PCB is ready for transportation to the insertion machine.

(2) The operator of the conveyor starts executing a transporting program. When the PCB is delivered to the right position and fixed, the operator can inform the operator of the insertion machine that the PCB is ready for insertion, or the latter can judge that through his/her own sensory system.

(3) The operator of the insertion machine can start executing an insertion program.

Operators’ roles in an office environment differ slightly from that on a shop floor. In an office environment, operators mainly deal with information transfers in large quantities between MEs. The executions of design and decision-making entities are not time-critical and the order of the executions is not strict and can also be concurrent. By comparison, on a shop floor operators are involved with not only timely executing machine programs in the right order and but also information transfers between shop floor MEs. However, the information
transfers are small in quantities but may be more frequent relative to that in an office environment. Overall, the roles of operators through the hierarchy can be considered as being:

(1) to collect from interrelated application programs within current stand-alone MEs the relevant information required for making new decisions or controlling machines and to update (insert, modify and delete) the information collected in an information repository coupled with the particular decision-making or control functions (i.e. application programs) to make this information available for decision making or control,

(2) to execute design, management decision-making, and control functions (i.e. application programs) in the right order.

It is important to remember that concurrent activities will occur and methods of synchronising events to satisfy the precedence constraints must be enabled.

4.4 Definition of CIM

Having analysed the composition of contemporary or stand-alone MEs, their interactions and the role of operators in manually integrated systems, a formal definition of CIM will be offered. Currently, there is no agreed definition of CIM (see chapter 2) just as for any new technology. In the author's opinion, what makes CIM different from "islands of automation" as characterized by CAD/CAM is the automation of the functions performed by operators as identified above. Therefore, from the viewpoint of integration, a CIM system can be defined as being a manufacturing system in which all application programs can automatically exchange information via information repositories and cooperate with each other whenever and wherever necessary to achieve higher level production flexibility and efficiency. A typical collection of contemporary
or stand-alone MEs are:
(1) financial accounting and other business functions,
(2) marketing,
(3) CAD,
(4) master production scheduling,
(5) process planning,
(6) material requirements planning,
(7) capacity planning,
(8) scheduling,
(9) quality control,
(10) shop control,
(11) cell control,
(12) control and monitoring of shop floor machines such as robots, PLCs, NCs, AGVs.

Compared with the definitions and explanations of CIM presented in Chapter 2, the author's definition of CIM is more precise and compact by emphasising, from the viewpoint of integration, automation for CIM in two aspects as identified above:
(1) information exchange between application programs via information repositories, and
(2) cooperation between application programs.

4.5 Summary

The functional requirements for systems integration tools have been determined by analysing the composition of contemporary or stand-alone MEs and their
interactions in manually integrated systems. These requirements will be used as the basis for formalising the design of a systems integration architecture in the next chapter.
5.1 Introduction

In this chapter, a systems integration architecture is to be formalised, based on the functional requirements for systems integration tools as identified in the last chapter. Available integration tools and methods will be assessed and finally the major project area will be pointed out.

5.2 Functional Requirements for Systems Integration Tools

It was explained in Chapter 4 that an enterprise can be considered as being composed of a variety of application programs and information repositories from the viewpoint of CIM systems integration. The operators' role is to transfer information between those application programs with possible information reformatting and to start application programs in the right order. Therefore, if operators are to be replaced by computerized systems integration tools, corresponding to the functions performed by operators, the collective functional requirements of these integration tools are:

(1) to ensure reliable and efficient enterprise-wide digital data transfer among computer nodes,
(2) to reliably and efficiently manage the storage of information generated by distributed application programs and the supply of information required by other distributed application programs,
(3) to automatically activate and coordinate distributed application programs according to pre-defined precedence constraints.

Requirement (1) is fundamental for meeting requirements (2) and (3) because it specifies the basic need for data communication links between computer nodes.

5.3 Systems Integration Architecture

The functional requirements for systems integration tools determine the systems integration architecture. Corresponding to the three functional requirements specified above, three systems are proposed to comprise systems integration architecture, viz:

(1) Digital Data Transfer System (DOTS),
(2) Information Administration System (IAS),
(3) Application Administration System (AAS).

The DOTS is to ensure reliable and efficient data communication among networked computer nodes. The IAS is to manage information storage, retrieval and update efficiently and reliably, providing automatic information services for all application programs and operators whenever and wherever necessary. The AAS is to automatically activate and coordinate application programs according to predefined precedence constraints. The relationships among the DOTS, IAS, AAS, application programs and operators, and CIM system programmers are:
(1) both the IAS and AAS utilize data communication services of DOTS,
(2) the IAS can directly provide application programs and operators with
information services rather than via the AAS,
(3) the AAS serves CIM system programmers with precedence definitions of
application programs and run-time activation and coordination of those
application programs, and
(4) there may not necessarily be direct interactions required between the IAS
and the AAS.

The ideal systems integration architecture proposed is shown in figure 5.1 (The
formal term "database" is used in figure 5.1 to replace the informal term
"information repository" for the reason which will be explained later in the
sub-section 5.4.4). The formalisation of the systems integration architecture
presented above was achieved jointly with other members of the L.U.T. Systems
Integration Group, which was based on the "three architecture" approach
proposed by Weston, et.al.. This formalisation provided support, substantiation
and enhancement to the "three architecture" approach by analysing, from the
viewpoint of integration, the composition of contemporary or stand-alone MEs
and their interactions in manually integrated systems. Based on the analyses,
more precise architectural specification was also attempted as presented above.

By dividing the complex CIM systems integration problem into the design and
implementation of three systems as stated above, major advantages accrue as
follows:
(1) CIM systems integration becomes conceptually clearer and more manageable.
(2) The CIM systems integration problem is decomposed so that expertise can
be brought to bear on each area, viz: the knowledge from the manufacturing
community on the AAS and the formulation of the IAS and DDTS by the
database (which will be explained later) and digital communication

-72-
FIGURE 5.1 SYSTEMS INTEGRATION ARCHITECTURE
(3) The decomposition allows much greater flexibility because different mechanisms can be developed to realise each of the three systems without affecting the other two systems.

5.4 Assessment of Available Integration Tools and Methods

5.4.1 Introduction

With reference to the systems integration architecture previously specified by the author, current available systems integration tools and methods will be examined and assessed. This discussion will facilitate the identification of database management technology as one of the most important integration tools.

5.4.2 The Role of ISO/OSI MAP/TOP

ISO/OSI MAP/TOP compatible computers and shop floor machines will become widely available in the foreseeable future. Already, many companies are including MAP and TOP implementations of the ISO/OSI reference model within their products. This is achieved by using a variety of single board computers which provide MAP and TOP connection to various parallel bus systems (e.g. Multibus, VMEbus and PCbus) [PAR88]. The MAP/TOP communication facilities can be incorporated in the backplane of many computer systems (e.g. for SUN workstations and IBM PC ATs) as a standard communication engine. MAP or TOP implementations of the ISO/OSI reference model can provide "end to end" communication in a multi-vendor environment with a certain level of decoupling of the communication function being possible.
Let us examine this statement in greater detail by considering the functionality of the highest layer (application layer) of MAP/TOP version 3.0 [GM87] (see figure 2.6).

A. FTAM

Both MAP and TOP have network-wide FTAM (File Transfer, Access and Management) facility, which is a significant step forward for CIM systems integration. With MAP/TOP FTAM, data files can be transferred from one make of computer or computer-controlled device to other makes. The contents of data files can be read from and written to, or these data files can be renamed and deleted, from any remote computer or computer-controlled device connected to the network. Whether in an office environment or on a shop floor, the three functional requirements previously identified must be met to achieve systems integration. FTAM certainly can neither activate and coordinate application programs, nor manage network-wide information storage and retrieval required by application programs. Nevertheless, FTAM facilitates file communication, thus providing a standard network-wide digital data transfer mechanism for DDTS.

B. MHS

Another service defined in the latest TOP version 3.0 is MHS (Message Handling System) which is really to accomplish the network-wide service of electronic mail [PUR87] [VER86].

C. MMS
MMS (Manufacturing Messaging Services) is an application layer service which is unique to MAP version 3.0. It can be viewed as a high level network command language [SIS87], which can facilitate interconnection of shop floor machines into a working cell. MMS utilizes the concept of Virtual Manufacturing Devices (VMDs) which provides a consistent view of the resources and functionalities of the real world devices such as robots, PLCs, NCs etc. [EIA87]. As mentioned in Chapter 2, companion standards are being appended to MMS. The companion standards will provide the functionality mapping between VMDs and real world devices [EIA87]. As far as MMS users are concerned, if MMS compatible machines such as robots, NCs, AGVs and computers are connected into a manufacturing or assembly cell, the functionalities available at these devices and computers can be accessed remotely and hence integrated [SIS87]. MMS comprises 81 services which are grouped into 10 "clauses" [EIA87]:

1. context management services,
2. VMD support services,
3. domain management services,
4. program invocation services,
5. variable access services,
6. semaphore management services,
7. operator communication services,
8. event management services,
9. journal management services,
10. file management services.

Being a network command language, MMS possesses certain features of high level computer languages as well as the functions specifically required in a
network environment. Variable access services, operator communication services and file management services are common functions of high level languages applied in a network environment. Due to the distributed nature of a network, functions such as context management services and VMD support services are necessary to allow MMS users to establish a dialogue, to enquire the status of functional objects residing in server MMS VMDs and to conclude a dialogue. Semaphore management services can facilitate sequential access of resources and functions when a number of MMS users are contesting the use of resources (e.g. controlling an AGV). Only does the MMS user who has held the semaphore have the right to use the resources controlled by the semaphore. Journal management services can be used to track and record activities and events which have happened, and this will be valuable for debugging application programs and recovering from break downs of machines and computers.

Among the functions of MMS, domain management services, program invocation services and event management services appear to be being advanced in the right direction so that the activation and coordination problems of application programs can be tackled. Program invocation services mean that there is a possibility to carry out network-wide remote activation of application programs (e.g. insert resistor A1001 into board 124). Domain management services and event management functions allow the definition of event conditions (e.g. the conditions for activating application programs) and actions. A machine so programmed can report information to a host computer when designated events occur, thus making it easy to realize the activation and coordination of application programs and supporting task-oriented application programming.

The concept embodied in MMS can also be utilized in an office environment for
coordinating application programs in performing design and various management functions. Hence, it should be suggested that the concept embodied in MMS be also applied in any OSI network (including TOP) although the coordination of office application programs is not as time-critical as that on a shop floor. In comparison with the progress made through MMS to facilitate the activation and coordination of application programs to meet the functional requirement of AAS, MAP and TOP are less advanced for achieving the information administration requirement of IAS.

5.4.3 The Role of Manufacturing Information Format Standards

The situation is that many manufacturing information format standards are competing for more supporters, among which PDES/STEP will potentially emerge as an international standard (see Chapter 2). The rationale behind these standard initiatives as reviewed in the literature survey is to solve information interpretation problems economically. This is of vital importance in approaching "CIM systems integration" which will be elaborated upon in Chapter 7.

5.4.4 The Importance of Database Management Technology

MAP/TOP without MMS would be a standard network-wide digital data transfer mechanism. MMS does partially meet the requirements of activation and coordination of application programs whereas the functional requirement to efficiently and reliably manage network-wide information retrieval and update remains untackled. Fortunately, database management technology has been aimed at solving this problem (which was recognised by the author through database literature reading and conceptual thinking). Sadly, the importance of this
technology is not widely realized in CIM systems integration community, being reflected in that, to the author’s knowledge, there is a lack of literature in this respect. Here, the author would like to emphasise strongly again the importance of database management technology for CIM systems integration.

Having realised the importance of database management technology, it is first necessary to explain formally what is meant by a database, a database management system and a distributed database management system. Here definitions, favoured by the author, are offered for each of these terms and they are the slightly modified versions of definitions given in the book [MAY81], with "data" being replaced by "information".

"A database is a collection of stored information items organized in such a way that all user information requirements are satisfied by the database. In general there is only one copy of each information item although there may be controlled repetition of some information items" [MAY81]

"A database management system is a general-purpose set of programs that aid and control each user’s access to and use of a database for adding, modifying and retrieving information, and this includes facilities giving information independence, integrity and security." [MAY81]

In the author’s opinion, a distributed database management system can be defined as a network-wide database management system to control and manage several distributed databases.

The rapid development during the last two decades in database management
technology will have impacts on achieving distributed information management for CIM. To design and implement an information administration system based on database technology was the major concern of this project. This work complemented the work of other members of the L.U.T. Systems Integration Group, where their work has largely been aimed at deriving novel forms of the AAS based on a MAP implementation of the DDTS. In the following chapter, the current state-of-art of database technology will be reviewed and assessed. Manufacturing-specific characteristics of information requirements will be studied so that a Factory Information Administration System (FIAS) can be developed by taking advantages of available database management technology.

To achieve this, all information repositories contained within current stand-alone MEs should be decoupled from application programs. Furthermore, the interfaces between decoupled application programs and information repositories should be redesigned. This point will become clearer in Chapter 9 as methods for achieving information access from within an application program are exemplified. In the future, proper stand-alone commercial DBMSs ought to be utilized for achieving efficient and reliable administration of factory information resources. This point will be substantiated in the later chapters.

5.5 Summary

In this chapter, a systems integration architecture has been formalised, which is decomposed into three systems, viz:

(1) Digital Data Transfer System (DDTS),

(2) Information Administration System (IAS),

(3) Application Administration System (AAS).
Available integration tools and methods have also been assessed. FTAM of MAP/TOP version 3.0 can be used as a standard data transfer mechanism. To some extent, MMS of MAP version 3.0 can partially solve the activation and coordination problem for application programs. Particularly, there is no doubt that database management technology is most closely relevant in satisfying information management requirements. Meanwhile, manufacturing information format standards are very important for economically solving information interpretation problems. In the following chapters, the creation of efficient, reliable and cost-effective AASs for CIM will be the major concern.
CHAPTER 6
ARCHITECTURAL DESIGN OF A FACTORY INFORMATION ADMINISTRATION SYSTEM

6.1 Introduction

The author recommends that the architecture of future CIM systems integration should comprise the following three separate systems as described in the last chapter:

(1) Digital Data Transfer System (DDTS),
(2) Information Administration System (IAS),
(3) Application Administration System (AAS).

The author believes that the key to CIM systems integration is the efficient management of information resources through the creation of information administration systems. Currently information management is performed either manually or partially computer assisted. Services of standard computer networks (such as FTAM of MAP/TOP LANs) can be utilised as the basis of realising the efficient computerized management of information resources as they provide a network-wide digital data transfer mechanism. The system used to manage information resources will be referred to here as a Factory Information Administration System (FIAS).

In this chapter, the importance of harnessing Distributed DataBase Management System (DDBMS) technology is emphasized in the building of future FIASs.
After the primary issues of DDBMS technology are reviewed and characteristics of manufacturing environments are analysed, the architecture of a suitable FIAS will be proposed based on the use of contemporary methods in information technology.

6.2 Importance of Distributed Database Management Technology for CIM Systems Integration

Information is costly to generate and manage. Accurate and timely information supply to all application programs whenever necessary is the key to the success of Factory 2000. The importance of information is rapidly growing in manufacturing industry as well as in every day life. For CIM systems integration, one of the major goals is an efficient and reliable computerized management of information resources. Database management technology was conceived to tackle this problem as pointed out in Chapter 5. This section is to develop this idea further by proposing the suitable technology in database management for CIM systems integration.

There are essentially two approaches when realising information management for CIM systems integration, viz: a centralized database approach and a distributed database approach. The distributed database approach is favored for CIM systems integration over the centralized database approach for the following reasons:

1. Storage limitation of a single computer

A medium or large manufacturing company has an enormous amount of information which is mostly impossible to be stored on and controlled by a single computer even if it is a mainframe computer.
(2) existing distributed databases
In a typical decentralized manufacturing company, a number of DBMSs may already be in use. The distributed database approach is a natural way of integrating those databases.

(3) potential reduction in communication costs
With the distributed database approach, many application processes can access information stored locally if the system is properly designed, thus reducing communication costs.

(4) potential quicker responses to users' information requests
Common problems with the centralized database approach include the long request queues and long communication time involved. If the system is designed properly using the distributed database approach, these problems can be alleviated. In other words, the responsiveness of the system to users' information requests can be increased.

(5) potentially higher reliability and availability
If the system is designed properly including the appropriate use of information duplication, the distributed database approach offers better reliability and availability than the centralized database approach where the failure of the central node can mean complete failure of information supply. Such a situation may not be tolerated in "time critical" manufacturing application areas.

As will be explained in greater detail in section 6.6.1 and 6.6.2, only through decentralization favoured administration of a distributed database approach can the benefits described in (3) (4) (5) be realised. To the author's knowledge, the potential value of DDBMS in CIM, like that of database technology in general, is not widely realized in the manufacturing community, this fact being reflected in that there is a lack of literature concerning the exploitation of this technology
in the manufacturing environments. Only in "world leading" research and development projects such as the American NBS IMDAS [KRI87] and US Air Force IISS [AF87], has the technology begun to be applied in manufacturing environments. However, as this technology becomes better understood and readily available in commercial form, it will have a significant impact on manufacturing practice and efficiency.

6.3 Issues of Distributed Database Management

6.3.1 List of Issues

A distributed database approach has been proposed by the author in creating a FIAS for CIM systems integration in the last section. Subsequently, existing experimental DDBMSs can be studied to understand many problems associated with applying this technology. Still, no perfect answers to any of the problems exist. After an extensive survey of distributed database management technology [BRA82] [CER84] [LAR85] [MOH84] [LOS88], nine major issues were identified and are listed below:

1. global conceptual data models and information formats;
2. information manipulation and definition languages;
3. information distribution optimization;
4. concurrency control;
5. transaction decomposition and processing optimization;
6. reliability and crash recovery;
7. information integrity;
8. information security;
9. distribution of administration functions.
Here, the term "information" is used to replace the conventionally popular term "data" for the reason stated in section 2.6.2. Explanations will be offered in the following sections concerning the ongoing nine issues.

6.3.2 Data Models, Information Formats and Information Manipulation and Definition Languages

A distributed database management system is designed to manage information distributed among a number of heterogeneous DBMSs which have different data models, information formats, and information definition and manipulation languages. As will be explained in chapter 7, one potential solution to these problems is to use SQL (Structured Query Language) as a standard information manipulation and definition language, and EER (Extended Entity-Relationship) model as a standard data modelling method. A move towards standardization of manufacturing information formats could be enabled through the activities of the PDES/STEP initiative. The author would propose SQL and EER model as a standard information manipulation and definition language and a standard data modelling method, a potential solution to be promised by the PDES/STEP committee.

6.3.3 Information Distribution Optimization

From users' requirements concerning information retrievals, information duplication is desired to increase the system responsiveness as users can access information stored locally, rather than suffer the consequences of transmission delays in networks. On the other hand, the cost of updating and storing information copies can also be significant. The objective of optimal information
distribution is to minimize the overall costs of information retrieval, update and storage.

6.3.4 Concurrency Control

When a number of users concurrently update the same information items under a multi-user operating system, concurrency control is vital to prevent them from interfering with each other, thus preserving information correctness and consistency in databases. Without a functional concurrency control mechanism, one of the typical problems is the "lost update" problem. This is because any update based on the value in a database involve three steps:

(1) read the base value into its workspace;
(2) modify the base value in its workspace;
(3) write the new value back into the database.

For example, the initial inventory of part X is 100, and salesman A has sold 10 of part X and salesman B has sold 20 of part X. These two transactions are entered into the database concurrently. The whole process is depicted in figure 6.1, with the final result "80" in the database being wrong. We know that the correct result is "70". The wrong result is due to the loss of salesman A's update which was overwritten by salesman B's update. There are additional complexities of concurrency control for distributed databases. Basically, two categories of concurrency control mechanism exist: two phase locking based and timestamp ordering based approaches. Chapter 8 will be devoted to developing a new concurrency control mechanism for distributed databases.

6.3.5 Transaction Decomposition and Processing Optimization
FIGURE 6.1 LOST UPDATE PROBLEM WITHOUT CONCURRENCY CONTROL
Transaction decomposition and processing optimization deals with the optimal selection, among an enormous number of alternatives of transaction decompositions and access plans. Request response time and total processing time are two criteria for transaction decomposition and processing optimization. Intersite communication time delay has been considered as the only time delay by a considerable proportion of researchers for transaction decomposition and processing optimization. This approach is acceptable for WANs in which intersite transmission speed is orders of magnitude slower than data transfer speed between disk and memory. In contrast, intersite data transmission speed in LANs contained within confines of a factory site is closer to that of local disk and memory data transfer so that the time for local I/O and CPU operations cannot be neglected.

6.3.6 Reliability and Crash Recovery

Reliability can be defined as "a measure of the success with which the system conforms to some authoritative specification of its behavior ...., when the behavior deviates from that which is specified for it, this is called a failure" [RAN78]. The reliability of a system can be measured as Mean Time Between Failures (MTBF). Availability is another way to state reliability. Crash recovery is the process of restoring a failed system to its correct operational state. The shorter the process of establishing crash recovery, the higher the system reliability or availability. A completely reliable distributed database management system must ensure that the system is 100% reliable when computer system hardware (including communication links) and system software are 100% reliable. Furthermore, a reliable system must be capable of minimizing the effects on information users if computer system hardware or system software
fails. Reliability and crash recovery will be crucial for totally integrated factories of the future.

6.3.7 Information Integrity

DDBMS must provide the mechanisms to ensure, as far as possible, that the information in databases is valid by enforcing a set of integrity constraints in the system. The "intelligence" required to enforce integrity constraints depends largely on the data modelling method upon which the system is built. The Entity Relationship model offers more capabilities for capturing semantics than classical hierarchical, network, and relational data models. The evolution of data modelling methods is directed towards more semantics capturing. Semantics Associated Model* (SAM*) is one of the examples, but it is difficult to be implemented in DDBMS. (see chapter 7 for details). In fact, the EER model will be proposed as a standard data modelling method.

Integrity constraints can take various forms, for instance:

1. constraints on the values of information items (e.g. the number of machines should be an integer greater than zero),
2. constraints on the values of information items in the same record (e.g. an employee's age must not contradict with his/her birthday),
3. constraints on the values of information items among several records (e.g. an employee's salary must not be higher than his/her manager's salary).

6.3.8 Information Security

Information security refers to the protection of the database against unauthorized
access, whether intentional or accidental. This is more important for multi-user distributed database environments. Illegal penetration of databases can have various levels of effect on a company. Budget and salaries are sensitive information which should never be modified or even retrieved by illegal penetrators. Information security can usually be controlled by following methods [CER84] [EVE86] [PRA87]:

1. password
2. external schema
3. authorization rules
4. encryption.

Firstly, passwords are very commonly used in computer systems to check whether users are authorized ones. They are also the effective tools for security control in DDBMS. Secondly, one of the side functions of the external schema is to establish information security because a user can only access information defined by his/her external schema. If user A is not allowed to see employee B's salary, we simply do not include the field of employee B's salary in user A's external schema. Thirdly, authorization rules define who has a right to access (retrieve, update, delete and insert) information items and under what conditions. For example, John (subject) is allowed to delete (action) a component processing routing (object) for which no customers have ordered the component (constraint). Fourthly, encryption is any sort of transformation applied to information items prior to transmission or storage to protect information from unauthorized accesses through bypassing DDBMS security control. This is transparent to authorized users because information is encrypted before stored in databases and is decrypted before delivered to authorized users. Unauthorized users who attempt to bypass the DDBMS security control can only see encrypted information whose meaning is not known to them.
6.3.9 Distribution of Administration Functions

Administration functions can be distributed among the computer nodes. Three types of distribution are possible:

(1) centralized administration: all administration functions are installed on a central node;

(2) fully distributed administration: all administration functions are installed on all nodes;

(3) hybrid administration: some nodes have full administration functions whereas other nodes only have partial administration functions.

6.4 The Proposal of Integrated Information Distribution Optimization System

Information distribution optimization identified previously was considered to be one of the profitable sub-areas. File allocation optimization techniques [CHU69] [MAH76] were not generally applicable in information distribution optimization amongst distributed databases [CER87]. The work of information distribution amongst distributed databases was pioneered by Ceri et. al. [CER87]. However, their proposals [CER87] were made on the assumption of simple requests and a very rough estimation of update cost. In general, information distribution optimization has been treated in isolation with respect to transaction decomposition and processing optimization, and concurrency control. This method is important for understanding each problem in depth. Otherwise, the whole problem is too complicated to be manageable. As a matter of fact, there is no optimal solution to any of the problems. While continuous efforts are needed...
to tackle each individual problem, their interdependency should be studied so that steps towards overall optimisation of information distribution can be made.

Information distribution optimization certainly depends upon the way in which transaction decomposition and processing optimization is carried out and the way in which concurrency control is realized. If only information updates are considered, information duplication is not desirable, whereas frequent information retrievals necessitate local information duplication since remote information accesses are inefficient. Therefore, ideally decisions concerning information distribution should be made with respect to the integrated cost evaluation taking into account the accumulated cost of storage, transaction decomposition and processing optimization, and concurrency control. The integrated system to make the decision can be called Integrated Information Distribution Optimization System (IIDOS).

It is appropriate here to introduce conceptual ideas which could indicate future directions of research in this area. Figure 6.2 illustrates a possible overall architecture of an IIDOS. Given a list specifying users’ transactions, the IIDOS could take heuristic approach, starting with minimum concurrency control and storage costs based on a non-duplicated centralized database model. A second step might be to partition the centralized database into fragmented databases and store information items at the node at which they are most frequently used. The overall cost evaluation module in the IIDOS could activate separate cost evaluation modules for transaction decomposition and processing, concurrency control and information storage. Subsequently, the overall cost could be computed and used as a basis for improvement. Then, the IIDOS could duplicate certain information items on those nodes where they are used with the second
FIGURE 6.2 INTEGRATED INFORMATION DISTRIBUTION OPTIMIZATION SYSTEM
highest frequency. If the overall cost is reduced, it is recorded and the process is repeated until there are no cost reductions. Finally, the IIDOS generates a list specifying how information items are partitioned and duplicated, which may not be optimal as a heuristic approach is taken. However, the title "Integrated Information Distribution Optimization System" reflects the ideal objective to be pursued. The foregoing decision-making process could be carried out by utilising a knowledge-based shell or ultimately involving the use of an inference engine.

6.5 Characteristics of Manufacturing Environments

6.5.1 Hierarchical Product Realisation Structure from the viewpoint of organization

Before the architecture of a FIAS can be devised, it is necessary to analyse the characteristics of manufacturing environments. We know that an enterprise is typically organised as a hierarchy which can be variably divided into a number of levels [FUR83] [JON84] [O'GR86]. The number of levels is somewhat artificial. Here, the author proposes that a hierarchical product realisation structure of an enterprise be divided into five levels from the viewpoint of organisation: machine level, cell level, workshop level, company level and enterprise level (see figure 6.3). Commands and feedback are the common interactions between MEs both among different levels and at the same level. Following are examples of commands and feedback among the five levels. However, this is not intended to be taken rigidly, rather is an illustration of the concept of hierarchical product realisation structure of an enterprise from the viewpoint of organisation. What is more important is the concept of non-hierarchical or heterarchical product realisation structure of an enterprise
FIGURE 6.3 HIERARCHICAL PRODUCT REALISATION STRUCTURE OF AN ENTERPRISE
from the viewpoint of information processing, which is to be introduced in the next section.

At the machine level, processing, handling and assembly processes take place to transform raw materials into semi-finished components and then finished products. Quality testing, which may be achieved manually or through using computer aided testing machines, can also be considered to belong to the machine level. Many types of information are required by machines to accomplish the processing, handling, assembly and testing of products, for instance, machine programs, geometric information describing parts and work-to-lists. Information is also generated at the machine level, e.g. work-in-progress information.

At the cell level, the major task of a cell controller is to coordinate the activities of machines as specified in work-to-lists created for each machine at the workshop or factory level. Work-in-progress information is passed by a cell controller up in the hierarchy.

Material Requirements Planning (MRP) and short to medium term production scheduling are commonly carried out at the workshop or company level. Such activities commonly utilise information generated both at the enterprise level and the cell level. Examples of information used at these levels are master production schedules, part routings, bill of materials and work-in-progress. Works orders, purchase orders and work-to-lists are the major information types generated at the workshop and company level.

Entities at the enterprise level are responsible for making decisions such as
marketing, financial accounting, long term production planning, master production scheduling, product design and process planning. Enormous quantities of information are generated at this level from design, production plans, master schedules, process plans, to finance and so on. The entities at this level make use of information from the external world and status feedback from the company level to carry out strategic planning.

We have seen from the organizational viewpoint that decision-making in a company is hierarchical. Decisions made at various levels are used to generate commands to lower levels, thereby specifying the tasks which they perform. The tasks may involve decision making and the generation of commands to yet lower levels in the hierarchy. At the lowest level, machines and people carry out the producing functions in response to higher level commands and generate status feedback for the cell level (e.g. work-in-progress), so does every other level for its successive higher level.

6.5.2 Non-hierarchical (Heterarchical) Product Realisation Structure from the Viewpoint of Information Processing

If we consider an enterprise from the viewpoint of information processing, all application programs whether they are design modules, management modules or machines modules are information users and/or generators. Decisions and status feedback are information, so are product drawings, payrolls, or customers' orders. From the viewpoint of information processing for CIM systems integration, an enterprise can be considered to consist of interacting information users and/or generators (i.e. application programs) and information repositories as explained in Chapter 4. The differences between individual information users and/or generators lie in the contents of information involved and the frequency
with which information retrievals and updates occur. Generally, information accesses will be most time-critical at the machine level and become less time-critical as the organisational hierarchy is climbed. This statement is true for the information demands of application programs, but not for human operators no matter which level they are from. A few minutes delay in information supply causes operators’ frustration.

Therefore, from the viewpoint of information processing for CIM systems integration, an enterprise can be considered as non-hierarchical (heterarchical) (i.e. being composed of a collection of interacting information users and/or generators and information repositories). This concept is of vital importance for systems integration because it indicates an easier way to achieve CIM systems integration than from the organisational viewpoint of an enterprise by breaking the boundary between organisational hierarchical levels.

Theoretically, the whole of an enterprise can be integrated as far as information administration is concerned. However, in practice, this is usually neither manageable nor maintainable after integration. Autonomy is desirable for minimizing the effects on the future whole integrated enterprise by a failure of a computer or a machine. Reasonably speaking, a workshop can be suggested as an autonomous unit to be integrated. Nevertheless, this is open to debate.

6.6 Architectural Design of a Factory Information Administration System

6.6.1 Administration Approaches
As described earlier, the administration of distributed databases falls into one of three categories: centralized administration, fully distributed administration and
hybrid administration. The criteria by which their merits and drawbacks are assessed are efficiency, reliability and system cost. The use of centralized administration for distributed databases, just like the centralized database approach, suffers from poor efficiency and reliability. The system cost with centralized administration of distributed databases is lower than that with fully distributed and hybrid administration because duplication of functions is required for both fully distributed and hybrid administration approaches. Nevertheless, the system cost is not the major factor in CIM scenarios if we imagine that all users' transactions are queueing for processing at the central node, and a failure of the central node means complete shutdown of information services for all users. Clearly, centralized administration is not an acceptable long term solution to the administration of distributed databases for CIM in medium or large companies. Furthermore, as the cost of computing power decreases, the use of fully distributed administration or hybrid administration will become more attractive.

Compared with the fully distributed administration approach, the hybrid administration approach is more practical because certain personal computers and computer-controlled machines may not have sufficient processing power to accomplish full administration functions and there may be no need to install certain functions such as a transaction decomposer (whose function will be explained later) if all transactions are simple.

A number of experimental distributed database management systems have adopted fully distributed administration, such as homogeneous SDD-I, system R*, ENCOMPASS, DDM, distributed INGRES, POREL, SIRUS-DELTA and heterogeneous MULTIBASE, and DDTS [CER84] [LAR85] [MOH84]. Using
such an approach, higher efficiency and reliability can be potentially realised than with centralized administration and the advantages of site autonomy can accrue. However, most of the distributed systems listed are really prototypes which do not have full information administration functions. For example, system R* does not support information duplication function and heterogeneous MULTIBASE does not support information updates. To the author’s knowledge, only two experimental truly distributed database management systems suitable for manufacturing environments have been developed, namely IISS by the US Air Force [AF87] [ALT87] [BAR87] [BRA87] [MUS87] and IMDAS by the American NBS [KRI87]. Overall, IMDAS offers more advanced information administration functions than the fully distributed administration based IISS which does not support information updates [LAR87]. A hierarchical three level architecture was designed to form the basis of IMDAS, which is known as BDAS (Basic Data Administration System), DDAS (Distributed Data Administration System) and MDAS (Master Data Administration System) [KRI87]. Each node must have a BDAS installed whereas there is only one DDAS required for servicing a set of nodes. The installation of a MDAS depends on whether there are more than one DDAS to be coordinated. Functionally, MDAS is the same as DDAS which can perform almost all administration functions, such as transaction syntax checking, authorization control, transaction decomposition and processing optimization, concurrency control, crash recovery and information assembly. In contrast, BDAS merely translates queries, accomplishes information translation and achieves interfacing to local DBMSs [KRI87]. In other words, the IMDAS hierarchical three level architecture is essentially a centralized administration system rather than a balanced hierarchical administration approach. As a result, the efficiency and reliability of IMDAS and its suitability for CIM are questionable.
6.6.2 The Proposal of Decentralization Favoured Administration

Taking into consideration efficiency, reliability, system cost, processing power and differences in information requirements, a "decentralization favoured administration" approach (as one case of hybrid administration) is proposed for future Factory Information Administration Systems. "Decentralization" is followed by "favoured" to stress the point that, whenever possible, decentralized decision-making and control are adopted, and centralized decision-making and control are adopted only when necessary. For the administration of distributed databases, many decision-making and control functions can be decentralized, including transaction syntax checking, authorization control, transaction decomposition and processing optimization, crash recovery, and information assembly. In contrast, concurrency control should be centrally enforced to preserve information correctness and consistency in databases. The condition to guarantee information correctness and consistency is that all update requests for the same information items must be serialized, which means that concurrency control, by nature, is a centralized decision. Further explanation of concurrency control will be presented in Chapter 8. The author believes that one of the problems with some of the existing experimental distributed database management systems mentioned previously is over-decentralization of the decision-making for concurrency control. Examples include DDM and SIRIUS-DELTA which were based on the decentralized two phase locking approach [CER84] [LAR85]. For concurrency control, the need for updating efficiency conflicts with that for system reliability. Thus, a balanced approach is favored, and the concepts involved here will be elaborated on in Chapter 8.
6.6.3 Modular Specification of a Factory Information Administration System

We have considered the major issues governing the nature of administrating distributed databases, the mechanism of an Integrated Information Distribution Optimization System and the rationale behind the selection of decentralization favoured administration approach. Subsequently, a Factory Information Administration System is proposed to comprise 17 modules (see figure 6.4), viz:

(1) Process creation manager
* is responsible for optimal system process creation and deletion according to the nature of an operating system and users’ request loads.

(2) System coordinator
* activates the right module under the right conditions. The actions of all modules are coordinated by the system coordinator except the "process creation manager", the "embedded SQL precompiler", the "information access statistics monitor", "SQL translator" and the "information translator".

(3) Terminal server
* prompts a terminal user to issue a transaction;
* passes the transaction to the message router;
* displays the resulting information on a screen or a window as available.

(4) Embedded SQL precompiler
* precompiles SQL statements embedded in an application program into the host high level language statements used in writing the application program which functionally interface with the message router for transaction delivery and information collection.
FIGURE 6.4 ARCHITECTURE OF FACTORY INFORMATION ADMINISTRATION SYSTEM
(5) Message router
* is responsible for routing the right messages to right modules.

(6) Communication manager
* is responsible for reliable message delivery to and acquisition from the rest of the computer network.

(7) Schema manager
* manages schema configurations;
* services schema enquiries.

(8) Transaction decomposer and processing optimizer
* decomposes compound transactions into simple transactions;
* validates information integrity by interacting with the schema manager;
* optimizes transaction access path.

(9) Transaction execution manager
* manages run-time transaction processing.

(10) Security controller
* checks users' access rights to information items;
* rejects unauthorized users.

(11) Concurrency controller
* serializes concurrent conflicting update transactions.

(12) Crash recovery organizer
* restores incorrect database states into correct database states in the event of system crash.

(13) Information assembler
* assembles information or acknowledgement fragments into a complete information or acknowledgement set in the right format.
(14) Local DBMS front end
* activates a local DBMS to obtain information or to update information (through SQL translator, information translator and information access statistics monitor).

(15) Information access statistics monitor
* records information access statistics.

(16) SQL translator
* translates transactions in SQL into transactions in a local query format for non-SQL based DBMSs.

(17) Information translator
* translates information coded in local formats into information coded in a standard global format.

Any administration function above should be installed on any computer node if the processing power of the node is sufficient and there is a need for the administration function. In this decentralization favoured administration approach, information administration instances on all nodes cooperate with each other to process transactions. However, certain concurrency controller instances have full power to serialize concurrent conflicting update requests with the others acting in supporting roles (this point will be expanded in Chapter 8). The functions of each module have be described briefly. Detailed functional explanations of most modules will be presented in Chapter 9.

6.7 Summary

Distributed database approach rather than centralized database approach is of vital importance to the success of Factory 2000 because efficient and reliable
use of information resources cannot be achieved without it. The nine major issues of DDBMS have been identified for building a FIAS through review of DDBMS technology. Opportunities are there, especially with regard to issues (3) (4) (5) (see sub-section 6.3.1), for researchers to accomplish improvements using existing technologies in order to build a more efficient, reliable or even intelligent FIAS for CIM.

It is now an appropriate time to study the interdependency of information distribution optimization on transaction decomposition and processing optimization, and concurrency control. Conceptually, a heuristic approach can be taken to generate near-optimal information distribution based on the overall cost evaluation of information retrieval, update and storage.

A heterarchical view of the conventionally accepted organisational hierarchy in the product realisation structure is essential in deriving a generic approach to building a FIAS. Decentralization favoured administration of distributed databases has significant advantages over centralized administration, hierachical administration, and fully distributed administration in establishing efficient and reliable FIASs. A FIAS was proposed to comprise 17 modules as listed in sub-section 6.6.3. The architectural concepts and ideas proposed can provide a platform for advancing methods of utilising information resources in completely integrated factories (or enterprises) during the 21st century.

Having accomplished the architectural design of a FIAS, the following two chapters will be concentrating on discussing two specific problems, viz:

(1) disparity in manufacturing information formats,
(2) concurrency control.
CHAPTER 7
A PROPOSAL FOR A MANUFACTURING INFORMATION STANDARD

7.1 Introduction

"Disparity in manufacturing information formats" is a serious problem when designing a FIAS. This chapter is aimed at resolving the problem to some extent by proposing a standard manufacturing data modelling method, and a standard manufacturing information manipulation and definition language. Finally, the appropriate stage at which information translations are accomplished will be suggested.

7.2 Disparity in Manufacturing Information Formats

As previously stated, for CIM systems integration MEs can be viewed to consist of application programs and information repositories (i.e. DBMSs). In the marketplace, there currently exist a variety of MEs which have their own specific formats for representing information and can only understand information in particular formats. This is the problem of "disparity in manufacturing information formats", which raises serious challenges for information integration.

Four approaches can be taken to tackle the problem:
(1) to persuade every ME supplier to stick to the same model of MEs
(including DBMSs),
(2) to purchase all MEs from one vendor,
(3) to produce specialized interfaces for all communication pairs of MEs,
(4) to produce standard interfaces for all MEs to hide the underlying differences in achieving functions.

The first approach is not realistic because it is highly unlikely that all ME vendors would stick to the same models in attempting to gain competitive advantages. Purchasing MEs from one vendor, if ever possible, would be against users' interests since very often each ME has its own merits and drawbacks and no ME is the best under all circumstances. That means the second approach is not practical either. Therefore, co-existence of MEs of various types is inevitable and also is a productive way. It is evident that large companies are in multi-vendor environments with the co-existence of MEs of various types, as is an increasing trend in medium to small sized companies. As presented in Chapter 2, approach 4 has advantages over approach 3 to cope with the problem for the reason that less translators are required (from now on, the more specific term "translator" will be used instead of "interface"). The comparison of approach 3 and 4 will be presented in greater depth in the next section.

7.3 Rationale for Establishing a Manufacturing Information Standard

The opportunity with approach 4 rather than approach 3 given above to reduce the number of information translators is the driving rationale behind activities aimed at worldwide manufacturing information format standardization. Furthermore, in-depth understanding of the importance of a manufacturing information standard would help CIM systems integrators to visualize the long term benefits of using such a standard.
Let us consider this situation in great detail by analysing the typical topologies of information transmissions. Information is passed between application programs (i.e. MEs) via special MEs (i.e. DBMSs). No matter how complex the information transmissions within a company are, four basic topologies for information transmission can occur as depicted in figure 7.1. The reduction in the number of information translators achieved through using an "information standard approach" over a "specialised information translator approach" can be seen clearly for the topologies illustrated by c and d (see also figure 2.14). This is true even if the set of information sources and users is fixed. The choice of enabling technologies is a highly complex and dynamic process, in which new types of MEs with novel features and better performance characteristics are continuously emerging in the international marketplace. International competition determines that manufacturing companies must upgrade their CIM systems more rapidly and pursue optimal purchasing of MEs. Moreover, for investment reasons their implementation should be decoupled from the activities of ME vendors. It is highly desirable to have a wide and free choice of MEs. If the need for configurability (i.e. upgrading) is taken into account, even the case of "one to one" information transmission in a long term plan presents the information disparity problem similar to the one of fixed "many to many" information transmission. This is because many choices of MEs are available for either end of the information transmission. The "information standard approach" can minimise the variety of information translators required to guarantee a wider freedom of choice for users. For the other three information transmission topologies, the increased complexity level involved means that greater benefits can be gained from using an "information standard approach". It minimises not only the overall variety of information translators, but also the number of
FIGURE 7.1 TOPOLOGIES OF INFORMATION TRANSMISSIONS
information translators added or replaced each time when a CIM system is expanded or updated. The exception to the latter is the situation where the "many ends" are updated under the "one to many" information transmission topology.

The above benefits of using an "information standard approach" can be realised by every ME user who is willing to adopt the approach. Furthermore, the benefits can be multiplied if a number of users can cooperate to develop a common manufacturing information standard. In such a situation, the unit cost of each information translator can be significantly decreased because ME vendors can sell manufacturing information translators in quantity to the user community rather than one-off quantities to single users. The greater the number of co-operating users, the greater the benefits they each can obtain.

The quality of a manufacturing information standard plays an important role in determining the realisable benefits to users. A manufacturing information standard should be so designed that the skills required to translate from vendors' manufacturing information formats to the standard manufacturing information format can be minimised. A high quality manufacturing information standard should contain many features found in the majority manufacturing information formats of vendors' systems in order to reduce the costs involved in information translations.

In summary, the author believes that ME users can gain the following benefits through adopting an "information standard approach":

1) minimization of the overall variety of information translators which are needed to guarantee users' wider freedom of ME choice,
minimization of the number of information translators which are added or replaced every time when a CIM system is expanded or upgraded for information transmission topologies b, c, and d in figure 7.1 with an exception of the case where only the "many ends" of topology b are upgraded,

(3) significant reduction in the unit cost of every information translator if ME users and vendors can cooperate to develop a common manufacturing information standard,

(4) further benefits gained through the design of a high quality manufacturing information standard.

7.4 Current Situations and Principles for Developing a Manufacturing Information Standard

The literature survey, in which various manufacturing information standards were reviewed, clearly indicates that the majority of the ME users' community in North America (particularly those related to aerospace industries) have chosen to cooperate in developing a common manufacturing information standard called PDES (Product Data Exchange Standard), which is a logical successor to the more specific IGES. The users' cooperation by itself is remarkable as it has been achieved among 250 companies and organizations, some of which are even competitors [DON87]. The extension of this initiative to include European effort to form the PDES/STEP information standard is also a logical progression, which should ensure a wider acceptance and greater potential benefits. The ME user community can thus expect to be able to realise the first three benefits listed above. The fourth benefit can be realised by designing a high quality manufacturing information standard which contains many features found in the
majority manufacturing information formats of vendors' MEs during the intended
duration of the standard.

Information is viewed by people and stored in DBMSs in a certain conceptual
structure designated as a data model. A data model can be informally viewed as
being analogous to a natural language grammar. Just as understanding of a
grammar is of vital importance in pursuing natural language translations,
comprehension of a data model can help to accelerate the process of
manufacturing information translations. To put it in another way, the greater the
differences between a standard manufacturing data model and vendors' specific
manufacturing data models, the more difficult the implementation of information
translations.

Information in manufacturing can be divided into queries for retrieving and
updating manufacturing information and the resulting information for queries.
Before standardizing manufacturing information formats, a data model should be
standardized since it is as important as the grammar of a natural language. As
we shall see in the next section, the data modelling methods for databases have
being developed ever since the idea of DBMS was originated in the late 1950’s.
These data modelling methods are continuing to develop to capture more
explicitly the meaning of the real world descriptions (i.e. semantics). Similar
developments have occurred with regard to the database query languages for
information manipulation and definition. Since the PDES initiative was launched
in 1986, the selection or design of a standard manufacturing data model has
been a high priority exercise [MOF86]. The Peoria data model has been
temporarily adopted as the basis of the early development stage of a standard
data model [MOF86]. Unfortunately, due to the infancy of PDES, detailed
information about the Peoria data model has not been available to the author. Up to now, no claim has been made about a standard manufacturing information manipulation and definition language (i.e. a database query language), which the author believes is necessary.

The remainder of this chapter will be mainly devoted to surveying existing data models and database query languages, and suggesting a standard manufacturing data model and a standard manufacturing database query language in the context of supporting the PDES/STEP movement.

### 7.5 Suggestions Concerning a Manufacturing Information Standard

#### 7.5.1 Historical Development of Data Models

The review of data models cannot be completely separated from the survey of DBMSs because data models are part of DBMSs. According to Fry [FRY76], the idea of DBMSs can be tracked back to the late 1950's when the authors such as McGee [MCG59] discussed the success of "generalized routines". Fry considered that the DBMS development before 1976 might be divided into three somewhat overlapping periods: the early development prior to 1964, the establishment of families during the period 1964-1968, and the vendor/CODASYL developments from 1968-1976. In fact, the period prior to the year 1968 was characterized by embryonic ideas with regard to DBMSs and prototypes of DBMS-like systems. After 1968, two distinct groups of DBMSs emerged, namely hierarchical data model based IMS (Information Management System) and network data model based CODASYL/DBTG family [FRY76] [TAY76]. A hierarchical data model represents the users' views of information
as forming information hierarchies while a network data model can be used to model the real world as information networks. When a hierarchical data model is used, it is difficult to model the many-to-many relationships which can in contrast be naturally modelled by a network data model [DAT81]. DBMSs based on these two data models constitute quite a proportion of commercial DBMSs even now.

In 1970 Codd introduced the relational data model approach [COD70], which was/is believed to have advantages over both the hierarchical and the network approaches [DAT81] [PRA87]. It allows people to model the real world information as a collection of tables or relations, whereby being simple, flexible and powerful. A description of relation theory and examples of the three data modelling approaches can be found in the book [DAT81], hence they will not be considered in greater depth here. The earliest relational database products were NOMAD, MAGNUM and Query-By-Example which appeared between 1974 and 1978 while more recent innovations around 1980 such as INGRES and system R represented comprehensive relational prototype DBMSs [DAT81]. Some relational DBMS products, for example, SQL/DS, DBASEII, DBASEIII, and INGRES, emerged in the marketplace in 1980's [BON84]. Currently, the majority of commercial DBMSs are based on either hierachical, network or relational data model with the clear tendency for relational DBMSs to become increasingly popular [DAT81] [BON84].

The proposal of the Entity Relationship model in 1976 [CHE76] was a significant landmark in the evolution of data modelling methods as the ER model is considered to be the first use of semantic data models [CHE83a] [VER88]. Since then, the desire to capture more meanings for database design
has been driving the whole process of data modelling, leading to various extensions of the ER model [CHE83b] [CER83]. The ER model has advantages over the relational data model in that the former allows database designers to perceive and model the real world more naturally and meaningfully than the latter (note: the real world being composed of entities and relationships) [CHE76]. The DBMSs based on the ER model or its extensions can use the semantics to enforce a set of integrity rules to prevent the databases from invalid updates more securely than relational DBMSs. In fact, the hierarchical, network and relational data models are currently classified as classical ones [VER88]. In other words, current thinking implies that modern data models must possess the capability of semantics modelling.

SAM* (Semantics Associated Model *), which is composed of seven semantic associations (e.g. membership association, generalization association), was claimed to be suitable for modelling integrated manufacturing information [SU86]. Its extension, OSAM* (Object-oriented SAM*), is being used as the data modelling method within the IMDAS of the AMRF at the American NBS [KRI87]. However, SAM* or OSAM* include more constructs and concepts which must be learnt such that they are more difficult to implement than the ER model [VER88]. The ER based data model is now accepted as the most popular data model in database design [TEO82] [SPE85].

7.5.2 Suggestion for a Standard Manufacturing Data Model

There are advantages and disadvantages with standardization. We have been stressing the benefits of standardization. At the same time, we must see its disadvantages, i.e. it tends to freeze the development of technology, especially
when the technology is not mature such as data modelling methods. It is important to agree on the duration of a manufacturing information standard, to devise a standard to best meet the needs during the period chosen, and to plan and organise changes to a standard whenever necessary. This suggestion for a standard manufacturing data model is reasonably made for the duration of ten years from 1990 to 2000, which is certainly open to debate. In the author's opinion, it is hard to predict what the most popular data modelling method will be after 10 years and the period of 10 years is considered to be a "foreseeable" period. Up to now, five major types of data models have been developed: hierarchical data model, network data model, relational data model, entity relationship based model (including original ER model and its extensions), and semantics associated based model (SAM, SAM*, OSAM*).

Compared with relational data model, both hierarchical and network data models are less favored as they are inflexible in information structures and are losing commercial popularity, thus should be ruled out as candidates. Among the remaining three types of data models (relational, ER based, SAM based), SAM based models are the youngest and the most complex to understand and implement, leading to the author's prediction that they will not have commercial popularity in the short and medium term.

Now, consider the two remaining candidates: relational and ER based models. The relational data model has current commercial popularity, but this popularity may not last for long because it is weak in modelling semantics, such a capability is crucial in facilitating further relieving of users' information management burdens. In contrast, the ER model and its extensions are more natural and competent in capturing semantics. Furthermore the ER approach has
also inherited the merits of relational data model (i.e. based on set and relation theory), whereby they have gained widespread public acceptance and experimental implementation. Therefore, the author predicts that ER based data models will be commercially most popular between 1990 and 2000, and proposes a semantics Extended ER (EER) model as a standard manufacturing data model. Here the term semantics Extended ER model is used to designate generally a sort of Extended ER model which has enhanced semantic modelling capability over the original ER model proposed in the paper [CHE76].

7.5.3 Suggestion for a Standard Manufacturing Information Manipulation and Definition Language

A manufacturing information manipulation and definition language is also referred to as a manufacturing database query language. The reason for selecting the most appropriate manufacturing database query language as a standard is the same as for selecting the most suitable manufacturing data model as a standard, i.e. to reduce translation costs. Thus the methodology in selecting a standard manufacturing data model will be followed. A standard manufacturing database query language should demonstrate many of the features found in most database query languages. This standard will again be proposed for the period from 1990 to 2000.

Although there are three classical data models: hierarchical, network and relational data models on which most current DBMSs are based, over 100 query languages were reported as having been developed before 1981 [SAM81]. In the book [SAM81], query languages were classified into two groups: user-driven dialogues and system-driven dialogues. There are three modes found under a
user-driven dialogue: (1) the constrained language mode, (2) the natural language mode and (3) the other user-driven dialogue. System-driven dialogues also have three modes which are: (1) the line-by-line prompting mode, (2) the menu selection mode and (3) the form fill mode. One example of a DBMS supporting more than one mode is dBASE III which can be accessed by either menu selection or a batch query language corresponding to the constrained language mode [LIS85]. The access mode of the internal database of MICROSS PP&C package [KEW85] falls into the category of menu selection mode.

For genuine classical DBMSs, example query languages are SQL, DL/I, and DBTG for relational R, hierarchical IMS and network DBTG respectively [DAT81]. SQL was shown to be the simplest to use in that a SQL query is a subset of a DL/I query which in turn is a subset of a DBTG query for accessing the same information items [DAT81]. The popularity of SQL has grown along with relational DBMSs. A refined SQL became an ANSI standard database language in 1986 [ANSI86]. Although the underlying data model of a DBMS influences the complexity related to the implementation of a database query language [DAT81], the framework of SQL is also suitable for the ER model [CHE76] [MAR83], SAM* [SU86], OSAM* [KRI87], and CSM (Construct Solid Model) and BR (Boundary Representation) model for CAD [KEM87] because all these models have a common basis as that of a relational data model, which is set and relation theory.

Having proposed the use of the EER model as a standard manufacturing data model, the author proposes SQL as a corresponding standard manufacturing database query language (or called a standard manufacturing information manipulation and definition language). The conclusion is mainly due to the
author's observation that the EER modelling method is widely accepted and shares a common basis as that of a relational data model for which SQL was originally developed.

Following is an illustration of the suitability of SQL for expressing database queries with an ER based model. Here objects mean both entities and relationships.

Select (Names of attributes)
From (Names of information objects)
Where (search conditions for information object occurrences)

Update (names of information objects)
Set (values to be updated for attributes)
Where (search conditions for information object occurrences)

Insert into (names of information objects)
values (information contents for information object occurrences and their attributes)

Delete From (names of information objects)
Where (search conditions for information object occurrences).

7.6 PP&C Data Modelling and Information Manipulation
The specification of a standard manufacturing data model is as important as that of the grammar of a standard natural language (if ever such a standard natural language could be produced). The huge scope, enormous variety and complexity
of manufacturing information necessitate a step-by-step approach towards a complete standard manufacturing data model. The original ER model [CHE76] would form the foundation of the standard manufacturing data model (i.e. the EER model proposed by the author) whereas the degree of semantics extension would remain the subject of open debates and extensive studies. Therefore, based on Chen’s ER model, a PP&C ER model will be developed and hopefully will be refined with semantic extensions at the following stages of conceptual thinking and research investigation.

7.6.1 Modelling PP&C Information in Chen’s ER Model

At this stage, Chen’s ER model is used by the author to model the information relating to scheduling and work-in-progress for the specific MICROSS PP&C package [KEW85]. The report details of MICROSS are considered as the users’ requirements or views of information, providing the basis of the PP&C conceptual ER model. To conserve space without sacrificing clarity, certain selected report details are presented in Appendix B.

The PP&C ER modeling will proceed in four steps:

1. identification of PP&C entity sets;
2. identification of PP&C relationship sets;
3. identification of PP&C attributes for entity and relationship sets;
4. representation of the ER model for PP&C in an ER diagram.

An analysis of the report details relating to scheduling and work-in-progress has led to the proposal of the following entity sets, relationship sets and their associated attributes.
(1) Identification of PP&C Entity Sets

- customers set ..... which specify the people who order products;
- parts set .......... which cover single components, sub-assemblies and final products identified by part numbers;
- machines set ...... which are physical manufacturing equipment used to make products.
- WIP-status set ..... which record the events of WIP operation completion.

(2) Identification of PP&C Relationship Sets

- orders set ....... which are issued by customers to indicate the types, and quantity of products they want and when they want the products.
- BOM (Bill of Material) set .... which specify the types and unit quantities of child parts used in the production of a parent part.
- operations set ..... which specify how parts are made on machines.
- work-to-lists set...which are the result of scheduling operations relating to when and on which machines all the part operations should be performed.
- work-loads set .... which, as the name implies, specify loads of part operations for machines as a result of scheduling.
- overdue-orders set ... which are the orders late for delivery.

The author identifies that order set, operations set, and work-to-lists set are relationship sets as well as entity sets, thus they are called dual entity/relationship sets.

(3) Identification of the Attributes of Entity and Relationship Sets
In the following listing, the attributes are quoted in parentheses following the
names of entity and relationship sets. Entity and relationship sets with their
associated attributes below are called entity relations and relationship relations
respectively, which can be used to define the global integrated schema in a
FIAS as will be explained in sub-section 9.5.1 and 9.6.1.

customers (customer-name, customer-address).

parts (part-no, part-description).

machines (machine-no, machine-description, capacity, No-of-Units, definition,
  performance-factor, modification).

WIP-status (order-no, (part-no, (operation-no, whether-completed))).

orders (customer-name, order-no, order-description, part-no, part-quantity,
  due-date).

ROM (parent-part-no, child-part-no, quantity-used).

operations (part-no, operation-no, operation-description, machine-no, set-up-time,
  unit-processing-time, transit-time).

work-to-lists (order-no, part-no, part-quantity, operation-no, operation-description,
  unit-processing-time, machine-no, SSD [scheduled start date], SFD
  [scheduled finish date]).

work-loads (machine-no, week-nos, normal-capacity, planned-capacity,
  cumulated-capacity, load-based-upon-ESD, load-based-upon-LSD,

overdue-orders (order-no, part-no, part-description, part-quantity, delivery-date,
  earliest-possible, scheduled-possible, date-late, priority).

(4) Representation of a Subset of the PP&C Data Model in an ER Diagram
In the ER diagram, rectangular boxes stand for entity sets, while diamond boxes represent relationship sets. There are also dual entity/relationship sets which are represented by a dotted rectangular box encompassing a diamond box. A resulting ER diagram representing a subset of a PP&C model for scheduling and WIP information is depicted in figure 7.2.

7.6.2 Manipulating PP&C Information in SQL

As previously stated, MICROSS is a menu-driven package which is composed of a database and application programs (or decision making programs). In commercial form, manipulation (i.e. insertion, retrieval, updating and deletion) of PP&C information is realized manually by selecting menu options and the subsequent typing-in of information items where required. For instance, the steps to be followed when obtaining the work-to-list for work centre no. 101 are ...

310 ... D ... 101 ... 101, which have no meaning to people who have not been trained in using it. The need for a specific and apparently meaningless dialogue creates problems when work-to-lists have to be retrieved both from within application programs and remotely from other network nodes with no access to the menus. In contrast, SQL is not only "rich" in semantics but is also suitable for manipulating the PP&C information in an EER model. SQL-based queries can be quite easily mapped onto menu selections in systems such as MICROSS.

Let us examine a few examples to see how stepping through menu selections for MICROSS can be mapped onto SQL based queries. The sequence comprises English statement, MICROSS menu selection and SQL based query.
FIGURE 7.2 A SUBSET OF PPIC ER DIAGRAM
(1) **English statement:** display work-to-lists for work centre no. 101 on the screen.

**MICROSS menu selection:** 310 D 101 101.

**SQL based query:**

```sql
Select * From work-to-lists Where machine-no="101".
```

(2) **English statement:** obtain operation routing for part 200.

**MICROSS menu selection:** 150 4 200.

**SQL based query:**

```sql
Select * From operations Where part-no="200".
```

(3) **English statement:** insert operation no. 1 for part 300, whose operation description is milling 1, set up time is 2 mins, unit processing time is 3 mins, transit time is 1 min, and processing machine is machine 100.

**MICROSS menu selection:** 140 3 300 1 milling 1 2 3 1 100.

**SQL based query:**

```sql
Insert into operation Values (300, 1, milling-1, 2, 3, 1, 100).
```

(4) **English statement:** Delete a customer's order, whose name is Peter.

**MICROSS menu selection:** 145 1 Peter.

**SQL based query:**

```sql
Delete From orders Where customer-name ="Peter".
```

7.7 **The Appropriate Stage for Information Translations**
7.7.1 Introduction

There have been three perspectives of the need for information translations to tackle the problem of "disparity in manufacturing information formats":

(1) network communication perspective,
(2) distributed database management perspective,
(3) the perspective from direct information transfers between CAD/CAM databases.

The remainder of this section is to describe these three perspectives in detail and to suggest an appropriate stage at which information translations should be carried out to design a cost-effective FIAS.

7.7.2 Network Communication Perspective

The ISO/OSI seven layer reference model includes presentation layer (layer 6), the purpose of which is to deal with information translation, compression and encryption. According to the ISO/OSI concept [ZIM80], information translation is one of the primary functions to be carried out by presentation layer to ensure that users at one end of a network can understand messages from those users at the other ends of the network without worrying about foreign message formats. These messages encompass queries expressed in database access languages and resulting information. Thus, information translations were perceived to be carried out in presentation layer from network communication perspective by the computer network community.

7.7.3 Distributed Database Management Perspective
As defined in chapter 5, the function of a DDBMS is to manage a number of databases distributed among several network nodes. DDBMSs can be classified as being homogeneous or heterogeneous according to their capability for managing different DBMSs run under different computer hardware and operating systems. If all are the same, a homogeneous DDBMS will suffice. On the other hand, any difference in those three aspects necessitates the use of a heterogeneous DDBMS. This kind of distinction is technically strict. However, from the viewpoint of DDBMS designers, differences in computer hardware and operating systems can be ignored, and the real concern about heterogeneity relates to differences in the user interfaces to the individual DBMSs. That is to say, the complexity of heterogeneous DDBMS over homogeneous DDBMS results from the disparity in the conceptual data models, query formats and the resulting information formats. These problems can only be overcome by information translations as recognized in the database community [BRA82] [CER84]. Thus, the need for information translations was perceived by both the computer network community as stated in sub-section 7.7.2 and the database community.

7.7.4 The Perspective from Direct Information Transfers between CAD/CAM Databases

As more CAD/CAM databases are used, it is desirable and sometimes necessary to copy information from one database to another, for example, the porting of product drawing information from the databases of a host design system to those subcontractors's databases. The differences in information formats make it very difficult, if not impossible, for information to be ported electronically among
different CAD/CAM databases, resulting in costly retyping. Therefore, people in the manufacturing community recognised the need for information translations. Take IGES as an example, which was originally designed to tackle information disparity problems encountered in porting design information between different CAD systems through information translations to the standard IGES format.

7.7.5 Summary of Perspectives

The computer network communication community, the database community, and the manufacturing community have all perceived the need for information translations to tackle the problem of "disparity in information formats" from their own angles. When designing a cost-effective FIAS, those three communities are suggested to combine their efforts to work out the best way to solve the common problem. The following sub-section will present the author's suggestion of an appropriate stage at which information translations should be carried out.

7.7.6 Appropriate Stage for Information Translations

As we have seen in section 7.3, the use of an "information standard approach" is a more cost effective way to accomplish information translations than that of using a "specialized translator approach". Furthermore, as we shall see later, the selection of information translation stage affects the overall integration costs, i.e. whether from network communication perspective or from the distributed database management perspective.

First of all, the systems integration architecture presented in chapter 5 is redrawn
here in a modified form (see fig. 7.3) to aid in explaining the effect on the cost of implementing information translations at different stages. For simplicity, only the Information Administration System (IAS) will be presented. The working procedure of the proposed IAS for serving a user’s information request could be the following:

(1) an original IAS agent accepts a user’s information query,
(2) it interprets the query and looks up a schema dictionary to find out the addresses of the required information,
(3) it decomposes the query if the required information is distributed,
(4) it sends the decomposed sub-queries to the correct cooperative IAS agents on the correct nodes,
(5) the cooperative IAS agents deliver the sub-queries to the appropriate individual databases with reference to local schema dictionaries,
(6) the cooperative IAS agents collect the resulting information fragments from the individual databases and send them back to the original IAS agent who originally accepted the user’s query,
(7) the original IAS agent assembles the resulting information fragments into a single information unit and delivers this information to the requesting user.

The procedures described above illustrate that an IAS is directly involved with the interpretation and decomposition of users’ queries, and the subsequent assembly of resulting information fragments from the individual databases. If information translations take place before queries and resulting information are transmitted into the IAS, the IAS can be simply designed to understand and decompose users’ queries, and assemble resulting information in single standard format, thereby the cost of the IAS can be reduced significantly. However, if information translations are carried out within the presentation layer, the cost of
AP = APPLICATION PROCESS, DB = DATABASE
IAS = INFORMATION ADMINISTRATION SYSTEM

Figure 7.3 Distributed Database Access
information translations will not be decreased. Meanwhile, the IAS will be completely exposed to different formats in users' queries and resulting information, resulting in significantly higher levels of complexity of the IAS and hence higher costs involving the interpretation and decomposition of users' queries in various formats, and the assembly of resulting information in different styles. Thus, it is obviously beneficial for all MEs such as CAD/CAM systems and DBMSs to be provided with query translators (QT) and information translators (IT) which translate from vendor specific information formats to a standard manufacturing information format. Using such an approach, the IAS needs only to deal with a standard manufacturing database query language and a standard manufacturing data model and resulting information formats (see figure 7.4), which is a cheaper way of automating the management of information resources.

7.8 Summary

The rationale for having a manufacturing information standard has been explained in detail. The EER (Extended Entity Relationship) model and SQL (Structured Query Language) have been proposed as a standard manufacturing data modelling method, and a standard manufacturing information manipulation and definition language respectively. The presentation layer of the ISO/OSI network reference model is not suitable for accomplishing information translations. The appropriate stage at which information translations should be carried out has also been considered. It is recommended that query/information translators for all MEs are required to be produced, which is the stage seen from distributed database management perspective. These proposals will facilitate a cost-effective design of FIASs.
QT = QUERY TRANSLATOR, IT = INFORMATION TRANSLATOR
AP = APPLICATION PROCESS, DB = DATABASE
IAS = INFORMATION ADMINISTRATION SYSTEM

FIGURE 7.4 APPROPRIATE STAGE FOR INFORMATION TRANSLATIONS.
The next chapter will be devoted to discussing concurrency control for distributed databases which is one of the important sub-areas for designing efficient and reliable FIASs.
CHAPTER 8
ADVANCES IN CONCURRENCY CONTROL
FOR DISTRIBUTED DATABASES

8.1 Introduction

As has been discussed in Chapter 6, FIASs are based on the technology of distributed databases. Concurrency control for distributed databases is one of the important problems which must be resolved for developing efficient, reliable and cost-effective FIASs. Accuracy and timeliness are two vitally important attributes of useful information. The choice of concurrency control mechanisms in multi-user systems affects both accuracy and timeliness. Concurrency control can be defined as those activities to prevent users' information transactions from interfering with each other, i.e. to serialize concurrent conflicting transactions in order to preserve the validity of information.

Inefficient concurrency control mechanism can adversely affect the timeliness of information services. Many concurrency control mechanisms have been proposed, which are essentially based on either locking or timestamp ordering (T/O) [BER78] [CER84] [BER80] [LEU86] [WAN87] [THO79] [KUN81] [ESW76] [ROS78] [WAN88]. In this chapter, those mechanisms will be reviewed and compared qualitatively and a proposal will be made to provide a more efficient concurrency control mechanism.

8.2 Problems Without Concurrency Control Mechanisms
8.2.1 Introduction

In a multi-user DDBMS without the existence of a concurrency control mechanism, three problems can arise to invalidate information or supply wrong information to users, viz:

(1) "lost update" problem,
(2) "inconsistent retrieval" problem,
(3) "incorrect and inconsistent copy" problem.

The "lost update" problem [BER81] [CER84] [DAT86] [PRA87] is universally recognized by the database community, whereas the "inconsistent retrieval" problem is, to the author's knowledge, only explicitly pointed out by Bernstein [BER81] and Date [DAT86]. Those two problems are common ones whether centralized database or distributed databases are used. However, the third problem (i.e. the "incorrect and inconsistent copy" problem) can arise when using distributed databases, which has not been explicitly identified by others.

8.2.2 Lost Update Problem

The "lost update" problem has already been exemplified in Chapter 6. Here, for completeness, another example will be presented to illustrate the "lost update" problem which would occur without the inclusion of a concurrency control mechanism. Suppose that there are 100 wheels in a warehouse of a company. Consider the case where one person from workshop A retrieves 10 wheels from the warehouse, recording this transaction via terminal A, while simultaneously another person from workshop B obtains 5 wheels from the warehouse and records the occurrence of the request via terminal B. Figure 8.1 illustrates this.
FIGURE 8.1 LOST UPDATE PROBLEM
procedure. The final result stored in the database is 95 (not 85 as required) as transaction B will overwrite transaction A, i.e. update transaction A can be lost. Clearly, the existence of incorrect information as the result of the loss of an update request will degrade the quality of decisions based on it, resulting in potential catastrophic effect. In this case, an MRP decision maker could assume that there are 95 wheels in the warehouse and decide to make or purchase 10 wheels less than necessary. Ultimately the wheel shortage will be recognised and the delayed delivery of goods to customers can result. Obviously, the effects of the lost update problem on a company, can be far worse if it happens in regard to the financial accounting.

8.2.3 Inconsistent Retrieval Problem

Let us look at figure 8.2. Suppose that transaction X is summing up the scrappage of component A located in workshop W1, workshop W2 and workshop W3 which are 100, 200 and 300 respectively, and transaction Y is transferring a number of scrapped components (say 10 of component A) from workshop W3 to workshop W1. The result produced by transaction X is 590 which is wrong because transaction X retrieves the scrappage information when the database was temporarily inconsistent before completion of transaction Y. The inconsistent retrieval problem may also dramatically affect decision-making. Nevertheless, in this case no permanent corruption of information happens in the database.

8.2.4 The Incorrect and Inconsistent Copy problem

Information duplication is a normal state of affairs in distributed database environments. Let us look at a financial example, with original budget of
FIGURE 8.2 INCONSISTENT RETRIEVAL PROBLEM
$100000 being duplicated at both computer A and computer B (see figure 8.3). Suppose that transaction X increases the budget by $10000 in respect to buying new equipment and that subsequently another decision is made to increase the budget following a second decision of equipment purchasing by 15% (this second transaction Y being entered via computer B). Suppose that transaction X updates copy A first at computer A and then copy B at computer B. If transaction X is propagated to copy B after transaction Y has already updated copy B, the non-synchronized executions of transactions X and Y cause copy B to be incorrect so that the two copies assume an inconsistent state.

8.3 Review of Various Concurrency Control Mechanisms

8.3.1 Introduction

As we have just discussed, the lost update problem, the inconsistent retrieval problem and the incorrect & inconsistent copy problem have a serious effect on the validity of information and the subsequent decision-making by users. In other words, concurrency control is essential to resolving such problems. Although many concurrency control mechanisms have been proposed, they are mainly based on locking or T/O. Ten major concurrency control mechanisms are listed as follows:

(1) basic T/O [BER78] [CER84],
(2) multiversion T/O [BER80],
(3) conservative T/O [BER78] [CER84],
(4) multidimensional T/O [LEU86],
(5) precedence agreement method [WAN87],
(6) majority consensus approach [THO79],
<table>
<thead>
<tr>
<th></th>
<th>COPY A</th>
<th>COPY B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORIGINAL BUDGET</td>
<td>$100000</td>
<td>$100000</td>
</tr>
<tr>
<td>AFTER TRANSACTION A</td>
<td>$110000</td>
<td>AFTER TRANSACTION B</td>
</tr>
<tr>
<td>AFTER TRANSACTION A+B</td>
<td>$125000</td>
<td>AFTER TRANSACTION A+B</td>
</tr>
</tbody>
</table>

**FIGURE 8.3 INCONSISTENT AND INCORRECT COPY PROBLEM**
(7) optimistic approach [KUN81],
(8) two phase locking (2PL) [ESW76],
(9) timestamp based deadlock prevention [ROS78] [CER84],
(10) unified concurrency control [WAN88].

Among them, 2PL is the most widely implemented [CER84] [DAT86] although conservative T/O was implemented in SDD-1 [BER78]. To the author’s knowledge, the remaining mechanisms have not been implemented in real systems. It is not appropriate here to describe each of these mechanisms in detail. Only the basic T/O and 2PL methods will be explained in depth as the other mechanisms are essentially variations of 2PL and the basic T/O. However, key features of the other mechanisms are highlighted. References are also included for readers wishing to gain an in-depth understanding.

8.3.2 Basic Timestamp Ordering (T/O) [BER78] [CER84]

When a transaction is entered into a system, it is assigned a timestamp. The timestamp for every transaction is unique so that all concurrent conflicting transactions can be executed in the timestamp order. If a transaction is aborted, it is assigned a larger timestamp before being reprocessed. Each information item in a database carries the timestamp of the last transaction which read or wrote the item. The basic T/O mechanism is listed as follows:

Step 1. Each transaction receives a timestamp when it is initiated at its site of origin.

Step 2. Each read or write operation, required by a transaction, carries the timestamp of the transaction.

Step 3. For each information item X, the largest timestamp of a read operation
and the largest timestamp of a write operation are recorded; they will be designated as $RTM(X)$ and $WTM(X)$.

Step 4. Let $TS$ be the timestamp of a prewrite operation $P_i$ on information item $X$. If $TS < RTM(X)$ or $TS < WTM(X)$, the operation is rejected and the issuing transaction is restarted; otherwise, a prewrite $P_i$ and its timestamp $TS$ are buffered.

Step 5. Let $TS$ be the timestamp of a read operation $R_i$ on information item $X$. If $TS < WTM(X)$, the operation is rejected. However, if $TS > WTM(X)$, $R_i$ is executed only if there is no prewrite operation $P(X)$ pending on information item $X$ having a timestamp $TS(P) < TS$. If there is one (or more) prewrite operation $P(X)$ with $TS(P) < TS$, $R_i$ is buffered until the transaction which has issued $P(X)$ commits. The reason why $R_i$ is buffered is that the write operation $W(X)$ corresponding to the prewrite $P(X)$ cannot be rejected. Therefore, we must avoid $TS(W) < RTM(X)$, but $TS(W) = TS(P)$, because they are issued by the same transaction. We must avoid applying $R_i$ since the value of $RTM(X)$ would be set equal to the value of $TS$, thus making $W(X)$ impossible. The read operation $R_i$ will be executed and eliminated from the buffer when no more prewrites with a smaller timestamp than $R_i$ are pending on $X$. After the read operation $R_i$ is executed, $RTM(X)$ is set to maximum($RTM(X), TS$).

Step 6. Let $TS$ be the timestamp of a write operation $W_i$ on information item $X$. This operation is never rejected. However, it is possibly buffered if there is a prewrite operation $P(X)$ with a timestamp $TS(P) < TS$, for the same reason stated for buffering read operations. $W_i$ will be executed and eliminated from the buffer when all prewrites with smaller timestamps have been eliminated from the buffer. After the write operation $W_i$ is executed, $WTM(X)$ is set to $TS$. 

-144-
In step 4, prewrites are issued by transactions instead of write operations and they are buffered and not applied directly to the database to meet the requirements of 2 phase commitment to ensure atomicity [CER84]. Only when the transaction is committed, are the corresponding write operations applied to the database. In this way, if the prewrite of a transaction has been accepted (or buffered), at the transaction commit, the corresponding writes will not be rejected. Buffering of an operation means that the operation is neither executed nor rejected, instead, it is recorded together with its timestamp for subsequent execution, and it is ensured that this execution will be possible at later time.

Merits

As timestamps are unique, according to Steps 4, 5 and 6, transactions are either rejected, executed, or buffered. Buffered transactions are guaranteed to be executed at later time because transactions in timestamp order cannot form a closed loop. There are no cases under which transactions wait for each other to form a loop as shown in figure 8.4. A deadlock in figure 8.4 occurs because transaction A waits for transaction B, B for C, C for D, D for E and transaction E waits for transaction A. In basic T/O, a transaction never hopelessly waits for another transaction. Instead, it either reads or writes an information item if it has a larger timestamp than the timestamp of the information item, or restarts with a larger timestamp if it has a smaller timestamp than the timestamp of the information item. Therefore, basic T/O is deadlock-free.

Drawbacks
FIGURE 8.4 EXAMPLE DEADLOCK
Drawbacks of basic T/O are:
(1) too many restarts,
(2) possible cyclic restarts.
All transactions which arrive late, to find that the information items have already been accessed by transactions with larger timestamps, are rejected and restarted with even larger timestamps. There is a possibility that a number of transactions can be restarted on a cyclic basis without ever finishing. Suppose that both transactions A and B read and write information item X and at the beginning transaction A has a smaller timestamp than transaction B. Figure 8.5 depicts how transaction A and transaction B cause each other to be restarted cyclically. Cyclic restarts can also happen to some unfortunate transaction interfered by many other different transactions.

8.3.3 Multiversion T/O [BER80]

Basic T/O can be improved by employing the multiversion T/O concept [REE78]. For each information item, a set of R-timestamps (read timestamps) and a set of <W-timestamp (write timestamp), value> pairs called versions are maintained. The R-timestamps of information item X record the timestamps of all read operations which have ever read information item X; the versions record the timestamps of all write operations which have ever written into X along with the values written.

By using multiversions, read operations of all transactions need not be rejected and restarted even if they arrive late. Let us look at figure 8.6 in which a read operation R1(X) with timestamp 65 arrives after the information item X has been written into by a write operation with timestamp 70. Because

-147-
FIGURE 8.5 EXAMPLE CYCLIC RESTART
FIGURE 8.6 EXAMPLE MULTIVERSION TIMESTAMP
multiversions of write timestamps and values of X are maintained, R1(X) just needs to read the value of X with largest timestamp less than 65. In this example, the value to be read by R1(X) is V(n-1). However, write operations which arrive late would still be rejected and restarted.

Merits

Multiversion T/O has the same merits as basic T/O, i.e. deadlock-free.

Drawbacks

Multiversion T/O still suffers from too many restarts and possible cyclic restarts although the number of restarts is reduced because read operations of all transactions are never restarted. Obviously, more storage space is needed to record multiversion timestamps and values.

8.3.4 Conservative T/O [BER78] [CER84]

The invention of conservative T/O was intended to eliminate restarts. Read and write operations are never restarted. Instead, they are buffered until all the read and write operations with smaller timestamps have been executed. All read and write operations are executed in timestamp order. A concurrency controller at site i works in the following way:

(1) For a read operation R that arrives at site i: if there is certain write operation W buffered at site i such that $TS(R) > TS(W)$, then R is buffered until these write operations are executed.
(2) For a write operation $W$ that arrives at site $i$: if there is certain read operation $R$ buffered at site $i$ such that $TS(W) > TS(R)$, or there is some write operation $W'$ buffered at site $i$ such that $TS(W) > TS(W')$, then $W$ is buffered until these read and write operations are executed.

Merits

As stated at the beginning of the section, conservative T/O eliminates restarts and it is also deadlock-free because all operations are serialized in timestamp order.

Drawbacks

Unfortunately, conservative T/O suffers from low efficiency. It serializes all read and write operations even if they are not in conflict (i.e. they do not access the same information items). It is not an exaggeration to say that conservative T/O is too conservative. To increase the degree of concurrency, transaction classes were introduced to allow transactions in different classes to be executed concurrently. This requires that each transaction predeclares all read and write information items to be accessed. In most cases, however, it is not possible to know in advance exactly what information items are to be accessed. Any way, the improvement of efficiency in this way is seen as minor.

8.3.5 Multidimensional T/O [LEU86]

Again, multidimensional T/O was proposed to improve basic T/O by reducing the number of restarts. In multidimensional T/O, each transaction has a
timestamp vector rather than only one timestamp. Timestamp items in a
timestamp vector are assigned dynamically based on dynamic validation of
conflicting operations of transactions. Two transactions can be assigned an
equal timestamp, stored in a timestamp vector, if all the information available
about the information items to be accessed indicates that both transactions are
not yet in conflict.

Merits

Multidimensional T/O inherits the merits of basic T/O, i.e. deadlock-free. To
some extent, the number of restarts is decreased by using dynamic validation of
conflicting operations and dynamic assignment of timestamps.

Drawbacks

Multidimensional T/O reduces the number of restarts due to the late arrival of
operations. However, it still cannot prevent operations arriving in timestamp
order from interfering with each other. Once the timestamp relationship is
decided between a set of transactions, they are fixed just like in a basic T/O
except that this decision is made at a later time. Thus, multidimensional T/O still
involves an enormous number of restarts and possible cyclic restarts. Timestamp
vectors occupy more storage space than a one dimensional timestamp in basic
T/O.

8.3.6 Precedence Agreement (PA) Method [WAN87]

Aimed at improving basic T/O, PA method was proposed to eliminate restarts. It
was claimed that it is deadlock-free as well as restart-free. Let us refer back to Step 4 and 5 of basic T/O. Read and write operations of late arrival are rejected and restarted to enable serialization of transactions. In PA method, these read and write operations are not restarted, but they will be assigned larger timestamps and executed subsequently without noticing the fact that some other operations have already interfered in the middle of execution. This results in the loss of some write operations. Therefore, PA method is not a functional concurrency control mechanism. Figure 8.7 illustrates how a write operation is lost.

8.3.7 Majority Consensus Approach [THO79]

The majority consensus approach was proposed by Thomas [THM79] to achieve concurrency control for fully duplicated databases. The basic idea is described as follows.

Every transaction is executed to its completion on its local copy of the database which is designated as an update list (U). A update list is submitted to every node and a voting process initiated. Updates are copied into the real database only if the majority of nodes have voted "yes". Otherwise, the transaction which issued the update operations is rejected and restarted with a larger timestamp. Further details are not intended to be given here. However, it is useful to examine the voting procedure which is described as follows:

1. Compare the timestamps of the read set of U with the corresponding timestamps in the database.
2. If they are not equal, vote "no".
3. If they are equal but there is a pending update list U' that has
FIGURE 8.7 LOST UPDATE IN PA METHOD
conflicting updates and a larger timestamp than $U$, vote "no".

4. If they are equal but there is a pending update list $U'$ that has conflicting updates and a smaller timestamp than $U$, defer the decision.

5. Otherwise, vote "yes".

In order to avoid a deadlock, a transaction should not be allowed to wait for a younger transaction as well as an older transaction. This is guaranteed by rules 3 and 4 as update operations with larger timestamps than pending updates are rejected or deferred.

**Merits**

Majority consensus approach is deadlock-free and resilient to node failure because the mechanism works on the basis of voting by operational nodes.

**Drawbacks**

The most serious problem is poor efficiency since the voting rule can lead to the occurrence of numerous restarts. Conflicting transactions can interfere with each other, just like in the basic T/O, thus leading to a restart problem at least as serious as with the basic T/O. Moreover, complete processing of transactions before voting for acceptance or rejection (in contrast to making decisions before processing in basic T/O) contributes to more expensive restarts than those with basic T/O in terms of consumption of processing power.

**8.3.8 Optimistic Approach [KUN81]**
Optimistic approach works in a similar way to that of the majority consensus approach except that a validation rule, which is different from the voting rule, is adopted. Timestamps for transactions are utilized to validate pending transactions, whereas in the majority consensus approach, timestamps for both transactions and information items are the basis of validation. All transactions must go through a read phase and a validation phase. Valid transactions proceed to a write phase and invalid ones are rejected and restarted. Figure 8.8 illustrates the three phases of this procedure.

Merits

Optimistic approach is deadlock-free for a single database.

Drawbacks

There are as many restarts as with the majority consensus approach and the restarts are more expensive than those of basic T/O for the same reason as with the majority consensus approach. Deadlock can occur if the use of the approach is extended to distributed databases. The optimistic approach is convenient for systems where the possibility of conflicts between transactions is low. Arguably, this is an advantageous condition for any other concurrency control mechanism.

8.3.9 Two Phase Locking (2PL) [ESW76]

The growing phase and the shrinking phase constitute the two phases in 2PL. In the growing phase, locks for information items are granted to a transaction
FIGURE 8.8 THREE PHASES OF TRANSACTION EXECUTION
gradually and none of the locks are released. After a transaction is committed or aborted, 2PL procedure enters into the shrinking phase during which the locks on the information items are gradually released and no more locks are requested by the transaction. 2PL is the most widely implemented concurrency control mechanism.

Merits

2PL is free of cyclic restarts and has much fewer restarts than timestamp-based mechanisms because only certain deadlocked transactions are chosen as victims to be aborted and restarted.

Drawbacks

Deadlock is the problem which the 2PL method faces. Deadlock is a situation where a circular waiting of transactions exists as shown in figure 8.4. Another example of deadlock can result as follows. Both transaction A and transaction B want to access information items X and Y. Transaction A locks information item X and transaction B locks information item Y at a similar instance in time (i.e. approximately simultaneously). Subsequently, transactions A and B proceed to lock information item Y and X respectively, which in this case have both been locked already. The result is a deadlock situation. A Wait-For-Graph (WFG) in figure 8.9 illustrates the deadlock situation (here a node or a circle is used to represent a transaction and a directed line stands for the fact that one transaction waits for another). Deadlocks block execution of transactions and take time to be detected, this being the main factor affecting efficiency in 2PL.
FIGURE 8.9 AN EXAMPLE WAIT_FOR_GRAPH (WFG)
8.3.10 Timestamp Based Deadlock Prevention [ROS78] [CER84]

Timestamp based deadlock prevention approach integrates the use of a timestamp method with the 2PL mechanism to prevent the occurrence of deadlocks because either the older transactions wait for the younger transactions or the younger ones wait for the older ones according to timestamps. Deadlocks can actually be prevented in two ways: nonpreemptive method and preemptive method.

The nonpreemptive method for deadlock prevention based on timestamps works as follows: if transaction Ti requests a lock on an information item which has already been locked by transaction Tj, Ti is permitted to wait only if Ti is older than Tj. If Ti is younger than Tj, Ti is aborted and restarted with the same timestamp. In short, conflict-causing older transactions wait for younger transactions and conflict-causing younger transactions are aborted and restarted.

A preemptive method functions differently. If transaction Ti requests a lock on an information item which has already been locked by Tj, Ti is permitted to wait only if it is younger than Tj, otherwise, Tj is aborted and the lock is granted to Ti. Conflict-causing younger transactions are allowed to wait for older transactions while conflict-causing older transactions preempt younger transactions, i.e. aborting and restarting younger ones. Nonpreemptive and preemptive method corresponds to the wait-die and wound-wait method described by Rosenkrantz [ROS78].

Merits
Both nonpreemptive method and preemptive method are deadlock-free.

drawbacks

Too many restarts occur with these deadlock prevention mechanisms. However, the restart problem is less severe than the one with the pure timestamp-based mechanisms for the reason that no transaction is allowed to interfere with other transactions in between the read and write operations corresponding to the same transaction on the same information item.

Comparision of Nonpreemptive Method and Preemptive Method

The nonpreemptive method makes conflict-causing older transactions wait for younger transactions and conflict-causing younger transactions be restarted. In contrast, preemptive method requires that conflict-causing younger transactions wait for older transactions and conflict-causing older transactions preempt younger transactions (i.e. younger transactions are aborted and restarted). By comparing the restart frequency, the preemptive method is superior to the nonpreemptive method. The chance of conflicts caused by younger transactions is higher than that of conflicts caused by older transactions because older transactions are likely to be the first to lock information items. Therefore, the nonpreemptive method leads to more frequent restarts than the preemptive method according to their different restart rules. Another reason is that restarted younger transactions are likely to be restarted using the nonpreemptive method before older transactions release locks whereas the restarting of younger transactions happens only once with the preemptive method. In conclusion, the preemptive method is more efficient.
8.3.11 Unified Concurrency Control [WAN88]

A unified concurrency control mechanism was suggested by Wang [WAN88]. It combines the basic T/O, 2PL, and precedence agreement method for the formation of more efficient concurrency control mechanism according to system parameters. The system parameters considered are as follows:

1. For 2PL transactions
   (a) average lock time if a request is not aborted
   (b) average lock time if a request is aborted
   (c) probability of abortion of a request due to deadlock
   (d) transaction arrival rate.

2. For basic T/O transactions
   (a) average lock time if a request is not aborted
   (b) average lock time if a request is aborted
   (c) probability of rejection of a read request
   (d) probability of rejection of a write request
   (e) transaction arrival rate.

3. For precedence agreement transactions
   (a) average lock time if a request is not backed off later
   (b) average lock time if a request is backed off later
   (c) probability of back off of a read request
   (d) probability of back off of a write request
   (e) transaction arrival rate.

The motive behind the unified concurrency control mechanism was that the result of a simulation [WAN87] indicated that the 2PL, basic T/O and PA method could outperform each other under different conditions. Specifically, the
simulation tested how the average transaction processing time \( S \) (as performance measure) is affected by transaction arrival rate \( R \) and the number of information items accessed by each transaction \( D \). 2PL performs well when \( R \) is low. When \( R \) is high, although the number of transactions directly involved in deadlocks does not increase very much, \( S \) goes up dramatically since more transactions are blocked by deadlocked transactions. For basic T/O, \( S \) grows steadily as \( R \) increases. It outperforms 2PL when \( R \) is high. However, as is also shown in [LIN83], the basic T/O becomes worse than both the 2PL and PA method as \( D \) increases. Apparently, this is due to the significant increase in restart probability. The PA method is a compromise between the 2PL and basic T/O methods. It performs similarly as 2PL when \( R \) is low and as basic T/O when \( R \) is high. When \( R \) is moderate, it outperforms both 2PL and basic T/O.

Merits

Assuming that the simulation results comparing the performance of the 2PL, basic T/O and PA methods are correct, the unified concurrency control mechanism would be a more efficient concurrency control mechanism than either 2PL, basic T/O or PA methods (when they are applied individually). However, as we will see later, the simulation results are questionable. Nevertheless, the unified concurrency control mechanism is a useful methodology when no single technique is the best under all operational conditions.

Drawbacks

Unfortunately, the PA method, as demonstrated in section 8.3.6, is not a
functional mechanism since it causes some updates to be lost. It is also questionable that the basic T/O can outperform 2PL even when the transaction arrival rate $R$ is high and the number of information items accessed by each transaction is small but larger than one. The probability of transaction restarts and cyclic restarts in basic T/O is extremely high even if the number of information items to be accessed by each transaction is two. Cyclic restarts consumes unlimited processing power. In contrast, deadlocks which could appear in 2PL can be detected by constructing WFGs and can be broken, which consumes less processing power. The author believes that it is highly unlikely that basic T/O can outperform 2PL in any practical situation.

8.4 Efficiency Evaluation of Various Concurrency Control Mechanisms

Having reviewed ten concurrency control mechanisms and considered their relative merits and drawbacks, one question arises immediately: which one is the most efficient? This section is aimed at answering this question on a qualitative basis.

Basic T/O does not prevent interference of transactions except during the period between the time of prewrite acceptance and the time of transaction commitment. As a result, too many transaction restarts and possible cyclic restarts lead to "poor" efficiency. Basic T/O can be categorized under a poor efficiency group. All efficiency evaluations will be based on a comparison with the basic T/O method.

Multiversion T/O slightly alleviates the problem of transaction restarts by eliminating any rejections and restarts of all read operations, which can, thus, be
classified as "less poor". Conservative T/O serializes the read and write operations of all transactions including non-conflicting operations, and hence essentially achieves no execution concurrency of transactions at all, leading to "poor" performance. The efficiency improvement in multidimensional T/O is realized through dynamic and as-late-as-possible assignment of timestamp elements in a timestamp vector rather than a single timestamp for each transaction. Concurrent conflicting transactions are serialized in timestamp vector order. However, this improvement is seen as being minor due to the fact that once the timestamp vector order is decided, it becomes fixed, meaning that subsequent read and write operations of out-of-order arrival are restarted and possibly on a cyclic basis. Multidimensional T/O can be labeled as "less poor" in terms of efficiency. The PA method "does not even work" because some updates could be lost (see section 8.3.6). The majority consensus and optimistic approaches are based on the same principle in that all transactions are executed to their completion on local copies of databases and applied to the real databases after conflict validation is passed, otherwise, rejected and restarted. Transaction interferences are not prevented by any means, thus the problem of transaction restarts and cyclic restarts are as serious as in basic T/O. The efficiency is even "poorer" than basic T/O since more processing power is wasted for executing transactions to their completion, which may later be restarted. 2PL has a much smaller number of restarts and no cyclic restarts, but has deadlock problem instead. Deadlocks can be detected and broken, which means that only limited time is wasted in comparison with cyclic restarts. A few restarts are only necessary for those transactions chosen for breaking deadlocks. Therefore, it is believed that 2PL is a "good" concurrency control mechanism. Timestamp based deadlock prevention mechanisms enforce locking on information items prior to accessing them. Transaction interference is prevented by locks. Only younger
conflicting transactions are restarted while all transactions are possibly restarted in basic T/O. In this case we can expect fewer restarts than with the basic T/O method. Another advantage is that it is deadlock-free. The preemptive method has fewer restarts than the nonpreemptive method (as explained in section 8.3.10). The number of restarts is higher than with 2PL. We can say that nonpreemptive method is "quite good" and preemptive method is "good". Although the methodology associated with unified concurrency control has some promising indications, the one proposed does not work because the PA method, as one of the three methods, is not workable, and it is questionable that basic T/O can outperform 2PL in any case.

The qualitative efficiency evaluation is summarized in figure 8.10. The grades for efficiency in an improving order are: "does not work", "poorer", "poor", "less poor", "quite good", "good".

8.5 Multiple Primary and Improved Two Phase Locking Concurrency Control Mechanism

The author believes that 2PL and preemptive deadlock prevention are better than the other eight mechanisms. In this section, problems associated with 2PL and preemptive deadlock prevention will be analysed in-depth and improvement will be suggested.

As stated previously, deadlock is the drawback with 2PL while preemptive deadlock prevention has the problem of many restarts. In fact deadlock is the extreme case of waiting, i.e. waiting forever. Waiting can happen among transactions which are not in deadlock. Waiting certainly has negative effects on
<table>
<thead>
<tr>
<th>CONCURRENCY CONTROL MECHANISMS</th>
<th>EFFICIENCY EVALUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASIC T/O</td>
<td>POOR</td>
</tr>
<tr>
<td>MULTIVERSION T/O</td>
<td>LESS POOR</td>
</tr>
<tr>
<td>CONSERVATIVE T/O</td>
<td>POOR</td>
</tr>
<tr>
<td>MULTIDIMENSIONAL T/O</td>
<td>LESS POOR</td>
</tr>
<tr>
<td>PRECEDENCE AGREEMENT METHOD</td>
<td>DOES NOT WORK</td>
</tr>
<tr>
<td>MAJORITY CONSENSUS APPROACH</td>
<td>POORER</td>
</tr>
<tr>
<td>OPTIMISTIC APPROACH</td>
<td>POORER</td>
</tr>
<tr>
<td>TWO PHASE LOCKING</td>
<td>GOOD</td>
</tr>
<tr>
<td>NONPREEMPTIVE DEADLOCK PREVENTION</td>
<td>QUITE GOOD</td>
</tr>
<tr>
<td>PREEMPTIVE DEADLOCK PREVENTION</td>
<td>GOOD</td>
</tr>
<tr>
<td>UNIFIED CONCURRENCY CONTROL</td>
<td>DOES NOT WORK</td>
</tr>
</tbody>
</table>

**FIGURE 8.10 EFFICIENCY EVALUATION OF VARIOUS CONCURRENCY CONTROL MECHANISMS**
efficiency. The problem is more serious in 2PL because any transaction can possibly wait for any other transactions. On the other hand, restarts occur more frequently in preemptive deadlock prevention.

One important point which should be stressed is that if any concurrency control mechanism attempts to eliminate "waiting" or "restart", it is bound to fail or not to allow any transaction processing concurrency. Restart-free conservative T/O employs "waiting" in an extreme way, leading to the loss of any transaction processing concurrency. Other T/O based concurrency control mechanisms mainly utilize "restarts" to control concurrent execution of transactions and the "waiting" is only harnessed to reduce the number of restarts, but not to stop cyclic restarts. Apparently, the possibility of "cyclic restarts" exists. In some sense, those T/O based mechanisms are not workable if cyclic restarts really happen. For a long time in the database community, "cyclic restarts" have been considered as being far less serious than "deadlocks". In the author's opinion, they are equally serious problems. Both "waiting" and "restart" mechanisms are used in preemptive deadlock prevention. Waiting time can be reduced by restarting certain transactions in a waiting chain. For example, suppose that transaction A is waiting for transaction B, transaction B for transaction C and so on until transaction K as shown in figure 8.11. Although a deadlock does not appear in this case, a lot of processing power is wasted on "waiting". If transaction F releases locks to transaction E and restarts afterwards, transactions E and K can be processed simultaneously, then transactions J and D and so on. The speed of processing those eight transactions approximately doubles when compared with the processing speed without restarting transaction F.

One way of improving preemptive deadlock prevention is to reduce "waiting".
FIGURE 8.11 AN EXAMPLE LONG WAITING CHAIN
WFGs must be built for identifying and breaking long waiting chains. This improvement has limits because the "restart" problem is still untouched. In order to improve preemptive deadlock prevention on a system wide scale (by reducing both "waiting" and "restart"), a "waiting" mechanism must be introduced to reduce the number of restarts and at the same time WFGs are used to break long "waiting" chains. The original preemptive deadlock prevention mechanism, which only allows younger transactions to wait for older transactions, is deadlock-free. Now, a potential improvement in efficiency on a system-wide scale requires that older transactions wait for younger transactions as well, which creates a deadlock problem. In other words, preemptive deadlock prevention mechanism is turned into 2PL if it is to be improved on a system wide scale. Consequently, it is logical that we can concentrate on improving 2PL.

2PL has the fewest restarts among the ten concurrency control mechanisms, but possibly has the longest waiting chains. We can break long waiting chains by restarting some transactions in the middle of chain as shown in figure 8.12. WFGs, which are used for breaking deadlocks, are also effective tools for breaking long waiting chains. A transaction chain involving three transactions is probably not a long chain. Suppose that each transaction requires one minute to complete and that transaction B does not release any locks requested by transaction A. In this case the processing sequence would be C--B--A and the total processing time for all three transactions would be (1+3)*3/2=6 minutes. If transaction B releases all locks required by transaction A, then transactions A and C can be processed concurrently, and the total processing time becomes (1 + 1 + restart time of transaction B + 1)=(3 + restart time of transaction B). Obviously, if the restart time of transaction B is less than 3 minutes, it is profitable to abort and break transaction B. What is considered to be a long
FIGURE 8.12 THREE TRANSACTION CHAIN
waiting chain will depend on a more detailed analysis of transaction processing time. Assuming that a five transaction chain can be considered to be a long transaction chain. Then, WFGs can be used to break one transaction for every three chained transactions because the third chained transaction is at the middle of five transactions. There may be another case under which multiple transactions are waiting for one transaction as depicted in figure 8.13. Execution concurrency can be increased by aborting the transaction waited for by the multiple transactions and granting locks to them in sequence.

Deadlocks must still be detected and broken locally as well as globally. This necessitates global WFGs and a global deadlock detector. A sensible scheme is to harness a hierarchical deadlock detection scheme such as that depicted in figure 8.14.

Essentially, the result of information distribution optimization is a hybrid information distribution with some information duplication. Two approaches have been suggested for updates: primary copy approach and update-all-copies approach [DAT86] [PRA87]. In the primary copy approach, all updates are first performed on the primary copy and subsequently propagated to all other copies. In this way, concurrency control is only necessary for the primary copy thereby facilitating high update efficiency. The weakness lies in that the failure of the primary node causes a failure for all update transactions. A primary copy approach has plus points in efficiency and minus points in reliability with regard to updating transactions. In contrast, the update-all-copies approach allows users to randomly issue update transactions to any possible node with the information copy, thereby

(i) causing more transactions to interfere with each other,
FIGURE 8.13 MULTIPLE TRANSACTION WAITING CASE
GDD = GLOBAL DEADLOCK DETECTOR
LDD = LOCAL DEADLOCK DETECTOR

FIGURE 8.14 HIERARCHICAL DEADLOCK DETECTION SCHEME
(ii) increasing the complexity for concurrency control and
(iii) degrading efficiency.

The update-all-copies approach can be failure tolerant or more reliable with respect to update requests if the scheme realizes concurrency control according to operational nodes. Therefore, a properly designed write-all-copies system has plus points in reliability and minus points in efficiency. Thus an ideal approach would be to balance the requirements of efficiency and reliability.

Assuming that every computer has a failure probability 10%, the failure probability of any two computers at the same time is 10%*10%=1%, which offers a considerable improvement in reliability. According to this principle, the multiple primary copy approach is suggested for multiple information copies. If under certain circumstances, the double primary copy approach is not reliable enough, the selection of a third primary copy can be considered. The principle is to balance reliability and efficiency. In line with the multiple primary copy approach, the global multiple deadlock detector approach can also be adopted to match equal degree of reliability and efficiency achieved by the multiple primary copy approach. Each deadlock detector has multiple backup agents which are best deployed on primary nodes. Hence the case cannot occur where a primary copy is available but none of its deadlock detector agents are functioning. This can be called compatible reliability.

To sum up, a novel concurrency control mechanism has been proposed by the author, which is suitable for use in FIASs based on distributed databases and can be named a "multiple primary and improved 2PL concurrency control mechanism".
8.6 Summary

It is essential to have a functional and efficient concurrency control mechanism for achieving information accuracy and timeliness. Various concurrency control mechanisms proposed in the past are essentially based on two techniques: locking and timestamp ordering (T/O). Those mechanisms can be associated with problems of waiting/deadlock and restart/cyclic-restart. A qualitative evaluation and comparison of those various mechanisms has led to the conclusion that 2PL is the best candidate for efficiency improvement. To improve 2PL and to further increase efficiency, not only deadlocks but also waiting chains have to be broken and certain transactions restarted. For a FIAS based on distributed database technology, if a balanced view of both reliability and efficiency is taken, the proposal is to adopt a "multiple primary and improved 2PL concurrency control mechanism".

In the next chapter, the implementation of a prototype FIAS will be described.
CHAPTER 9
A PROTOTYPE FACTORY INFORMATION ADMINISTRATION SYSTEM

9.1 Introduction

Having considered the architecture of a FIAS, the manufacturing information standardization and concurrency control for a FIAS, a detailed implementation of a prototype heterogeneous FIAS will be presented. This prototype FIAS has been used to integrate a relational DBMS (having an SQL interface) and a specific DBMS (having no SQL interface) to provide comprehensive information services with access transparency and system configurability.

9.2 Consideration of Project Facilities

Like other MEs, DBMSs available on the marketplace are of various types with their major differences being related to their use of differing database access languages and data modelling methods. For instance, they can be built upon hierarchical, network, relational and other data models, and differ in other aspects, such as concurrency control and crash recovery. These disparities offer major challenges for systems integrators where network-wide information sharing requirements exist. There are essentially four approaches which can be taken to meet those challenges:
(1) to persuade every DBMS vendor to stick to one DBMS model,
(2) to purchase all DBMSs from one vendor,
(3) to provide specialized translators for all pairs of DBMSs and other MEs in use,

(4) to produce standard translators for DBMSs to hide the underlying differences in achieving DBMS functions.

As discussed in Chapter 7, the only practical and productive method to tackle the disparity problem is the fourth approach in which standard translators are produced for each DBMS either by users or vendors.

The principle of accepting heterogeneity in DBMSs has a direct parallel in accepting necessary heterogeneity with respect to computer hardware. Based on a recognition of the necessary co-existence of various types of DBMSs and computer hardware, the requirements of a prototype factory information administration system were specified and implemented. With general industrial applicability in mind, the hardware and software facilities embodied in this prototype system were chosen to be representative in heterogeneity. An IBM PC AT with MS-DOS operating system and a SUN workstation with UNIX operating system (which are to represent single tasking personal computers, and multi-tasking mini-computers or multi-user mainframes respectively) were chosen as the hardware. The DBMSs chosen were a relational DBMS called INGRES [REL86] which runs on SUN workstations and a specific DBMS named MICROSS [KEW85] which runs on IBM PC ATs. For reasons of expediency (i.e. a MAP data link was not available at the time of creating the FIAS), the data link between these two computers was based on Kermit file transfer [CRU85a] [CRU85b] (see Appendix A). Figure 9.1 depicts the facilities chosen.

Two points need to be stated clearly about the facilities. The first point is that although MICROSS DBMS is a part of MICROSS PP&C package and not a
FIGURE 9.1 PROJECT FACILITIES
proper stand-alone commercial DBMS, it possesses typical characteristics of a commercial DBMS, i.e. it supports information storage, update and retrieval. This is done in an ad hoc language which is different to SQL. However, SQL is supported by INGRES. This heterogeneity in database access languages was a pre-requisite requirement in this study of the generic integration problem. Arguably, the ideal choice would be a proper stand-alone commercial DBMS with a non-SQL interface. Nevertheless, there is no doubt about the generality of ideas evolved through studying the chosen DBMSs. The second point is that the usage of Kermit file transfer does not restrict the generality of the ideas. Kermit file transfer provides a file transfer service between two computers which can be easily replaced with the FTAM service of MAP/TOP. This is because both are equivalent in terms of types of file transfer service, and only certain aspects of scope and performance are different. FTAM of MAP/TOP can guarantee file transfers on a network-wide basis rather than a point-point one and with higher speed than Kermit file transfer. However, this difference does not affect the generality of ideas developed either. In other words, the system results will not be specific system facility dependent, but generally applicable in a wide variety of CIM scenarios.

9.3 Functional Requirements of the Prototype System

An appreciation of the perceived functional requirements of the system will be required in understanding the chosen system architecture and mechanisms. Functional generality originates from users’ general requirements. Thus, first of all, information should be allowed to be distributed and duplicated to meet users’ general information distribution requirements. Secondly, information must be accessible through both terminals and from within application programs.
Thirdly, manipulation of information should be allowed through two categories of operations: retrieval and update (including insertion, deletion and modification). The operation of the system should be transparent to users with respect to how information is fragmented, whether information is duplicated and where information is stored. Fourthly, industrial companies (especially successful ones) will need to restructure their product realisation functions after a period of time, introducing new products, expanding or enhancing production facilities and possibly forming co-operative associations or other associations with other companies. Those changes necessitate the restructuring, creation and deletion of information. A change of users will lead to re-deployment of information among various DBMSs to enable efficiency and reliability of use. Thus, configurability of a factory information administration system is of vital importance for its durability and enabling transparency of use under changes (*).

Therefore, implementation of the following generic functions was considered as being necessary:

1. Support functions for information distribution and duplication,
2. Support information access functions from both terminals and within application programs,
3. Support functions to enable generic manipulation of information:
   3.1 Local retrievals in SQL

(*) The importance of configurability cannot be overstressed. Extremely large investments are placed in factory or enterprise information systems, requiring very large incremental investments in the face of change if inflexible information support facilities exist. A fundamental requirement of Factory 2000 will be configurability.
(3.2) remote retrievals in SQL
(3.3) compound retrievals in SQL
(3.4) concurrent compound retrievals in SQL
(3.5) local updates in SQL
(3.6) remote updates in SQL
(3.7) updates for data duplications in SQL
(3.8) concurrent simple updates in SQL,
(4) support functions to enable system configurability.
By "compound retrieval" it is meant that the retrieval requires information stored at more than one computer node or site.

9.4 System Architecture and Mechanisms

An example information distribution and duplication arrangement is depicted by figure 9.2, which allows the creation of a generic information distribution environment for implementing and evolving techniques for achieving the other three function classes. The system architecture is shown in figure 9.3, being composed of the following modules: system coordinator, communication manager, terminal server, message router, transaction decomposer, information assembler, schema manager, local DBMS front end, SQL translator and embedded SQL precompiler. Before the detailed functionality of each module is presented, the system roles of certain modules will be outlined to illustrate mechanisms for enabling the generic requirements discussed in the last section. The terminal server module interacts directly with terminal users while the embedded SQL precompiler provides capabilities for allowing information accesses within application programs. Transactions, whether from terminal users or from application processes, would be processed by the transaction decomposer
FIGURE 9.2 EXAMPLE INFORMATION DISTRIBUTION & DUPLICATION
FIGURE 9.3 PROTOTYPE FACTORY INFORMATION ADMINISTRATION SYSTEM
module and the role of the information assembler is to assemble resulting information and acknowledgement fragments. An "acknowledgement" message is the message for reporting the processing of a transaction. System configurability is realized through a schema manager.

9.4.1 System Coordinator

In the case of a single tasking operating system, the system coordinator coordinates the activities of the transaction decomposer, information assembler, schema manager, local DBMS front end and message router as well as the terminal server and communication manager. In comparison, in the case of multi-tasking operating system, the system coordinator does not control the terminal server and communication manager as they run as independent processes. The activation of a module is based on the existence of messages in its message queue. The approach is illustrated by the flow chart shown in figure 9.4(1) and 9.4(2).

9.4.2 Communication Manager

The communication manager is responsible for the reliable transmission of messages to and from the right nodes by abiding by certain communication protocols. In the prototype implementation, the communication manager (running on the SUN workstation) functions as a communication server whereas the communication manager instance on the IBM PC AT acts as a communication activator, which means that bi-directional message passing is activated solely by the communication manager on the IBM PC AT (depicted in figure 9.5). The communication managers in this implementation make use of Kermit file transfer
ACTIVATE TRANSACTION DECOMPOSER

ANY REQUEST FOR TRANSACTION DECOMPOSER

ACTIVATE LOCAL DBMS FRONT END

ANY REQUEST FOR LOCAL DBMS FRONT END

ACTIVATE INFORMATION ASSEMBLER

ANY INFORMATION OR ACKNOWLEDGE FOR INFORMATION ASSEMBLER

ACTIVATE SCHEMA MANAGER

ANY REQUEST FOR SCHEMA MANAGER

FIGURE 9.4(1) FLOW CHART OF SYSTEM COORDINATOR FOR MULTI-TASKING OPERATING SYSTEM
ACTIVATE COMMUNICATION MANAGER → ACTIVATE TERMINAL SERVER → ACTIVATE MESSAGE ROUTER

ANY REQUEST FOR TRANSACTION DECOMPOSER

ACTIVATE TRANSACTION DECOMPOSER

ANY REQUEST FOR LOCAL DBMS FRONT END

ACTIVATE LOCAL DBMS FRONT END

ANY INFORMATION OR ACKNOWLEDGE FOR INFORMATION ASSEMBLER

ACTIVATE INFORMATION ASSEMBLER

ANY REQUEST FOR SCHEMA MANAGER

ACTIVATE SCHEMA MANAGER

FIGURE 9.4(2) FLOW CHART OF SYSTEM COORDINATER FOR SINGLE TASKING OPERATING SYSTEM - 187 -
COLLECT MESSAGES FROM LOCAL QUEUES FOR OUTPUT

CALL NETWORK FILE TRANSFER

ENTRE SERVER MODE AND WAIT PASSIVELY FOR REQUESTS FROM THE OTHER COMMUNICATION MANAGER

SEND ALL MESSAGES

COLLECT ALL MESSAGES

QUIT THE SERVER

BACK

PUT ALL THE MESSAGES FROM REMOTE NODES INTO THE QUEUE FOR MESSAGE ROUTER

FOR MULTI-TASKING OPERATING SYSTEM

FINISH

COLLECT MESSAGES FROM LOCAL QUEUES FOR OUTPUT

CALL NETWORK FILE TRANSFER

FOR SINGLE TASKING OPERATING SYSTEM

FIGURE 9.5 FLOW CHART FOR COMMUNICATION MANAGER
services, the details of which can be found in Appendix A.

9.4.3 Terminal Server

The terminal server acts as an intermediary between the terminal users and message router. When running under the UNIX multi-tasking operating system, it accepts transactions from terminal users and assigns them to the "request pipe"(*) for message router to collect. Subsequently it "continuously", from the view of a programmer, checks the "reply pipe" for information or acknowledgements and displays the result on the screen when available. When running under a single tasking operating system, the mechanism is slightly different. The terminal server can only be a casual process, in other words, it cannot wait for requests and information by continuous polling. The flow chart (see figure 9.6 (1) (2)) illustrates these processes in some detail.

9.4.4 Message Router

The message router is responsible for gathering transactions and information from various sources or channels: i.e. from the terminal server, communication manager, application processes, schema manager, transaction decomposer, information assembler and local DBMS front end modules and routing these messages to the right channels. For example, initial transactions are routed to the transaction decomposer, decomposed transactions for remote nodes to the

(*) Although this "pipe mechanism" is UNIX dependent, it is believed that there are similar mechanisms in any other multi-tasking operating systems.
START

ANY REQUEST

Y

COLLECT REQUEST

PUT REQUEST INTO 'REQUEST PIPE'

ANY REPLY

N

Y

COLLECT INFORMATION OR ACKNOWLEDGEMENT FROM 'REPLY PIPE'

DISPLAY INFORMATION OR ACKNOWLEDGEMENT ON THE SCREEN

FIGURE 9.6(1) FLOW CHART FOR TERMINAL SERVER FOR MULTI-TASKING OPERATING SYSTEM
FIGURE 9.6(2) FLOW CHART FOR TERMINAL SERVER FOR SINGLE TASKING OPERATING SYSTEM
communication manager, unassembled information and acknowledgement fragments to information assembler and complete information or acknowledgements to the application processes or the terminal server. Figure 9.7 flow-charts this arrangement.

9.4.5 Transaction Decomposer

The transaction decomposer decomposes transactions into simple transactions by looking up the information schema provided by the schema manager. Simple transactions are those transactions which can be processed by single DBMS. The information which codes how transactions are decomposed is stored for the information assembler to assemble information fragments at a later time (see figure 9.8).

9.4.6 Information Assembler

The information assembler references transaction decomposing indications which are created by transaction decomposer and assembles all information or acknowledgement fragments. If information is complete for a given transaction, the system coordinator is informed that certain information or acknowledgements are complete and ready for shipment to users. If incompleteness of acknowledgement is due to information duplication, the information assembler issues the right transaction to the nodes concerned for the updating of other copies. The detailed procedure is illustrated by figure 9.9.

9.4.7 Schema Manager
I C!LL.ECT I£SSACES FROM TERMINAL ROUTE I£SSACES TO ROUTE I£SSACES TO
SfR/EII TEIlI1INAL SfR/EII INFOflIfATlCW.

IF THERE ARE ASSEIIBLER AI« FOR IT IF
1£IlE ARE C!ll.LE CT I£SSAGES AI« FOR IT
FROH APPliCATION PROCESSES

CCUECT I£SSAGES ROUTE I£SSACES TO ROUTE I£SSACES TO
FROH ·INFORHATlCW SO£IfA
ASSEItBlER twIAC£R
I IF THERE ARE
IF THERE ARE
AI« ( Ai« FOR IT

C!ll.LE CT I£SSAGES FROH lOCAL DBHS FRONT EHl
I
C!ll.LE CT I£SSAGES ROUTE I£SSACES TO
NOTIFY THE FROIt SCHEIlA SYSTEIf 

twIAC£R TRANSACTlCW COOROINATER

I DECOHPOSER IF
THERE ARE FINISH
SORT OUT WHICH AI« FOR IT
IlMESSAGE IS FOR WHICH NODULE

FIGURE 9.7 FLOW CHART FOR MESSAGE ROUTER
START

COLLECT A TRANSACTION

CHECK THE GLOBAL INTEGRATED SCHEMA TO SEE IF ALL THE TABLES IN THE TRANSACTION ARE VALID

ARE ALL THE TABLES VALID

Y

CHECK THE GLOBAL FRAGMENTED SCHEMA TO SEE HOW THE TABLES ARE FRAGMENTED AND VALIDATE THE SEARCH CONDITIONS

DECOMPOSE THE TRANSACTION INTO SIMPLE TRANSACTIONS BASED ON FRAGMENTATION & ALLOCATION AND ATTACH THE ADDRESSES TO THE SIMPLE TRANSACTIONS

STORE THEM IN QUEUE FOR MESSAGE ROUTER

N

REPORT THE INCORRECT TABLE(S)

REPORT THE ERROR CONDITIONS

ARE ALL SEARCH CONDITIONS VALID

Y

NOTIFY THE SYSTEM COORDINATOR

N

CHECK THE GLOBAL ALLOCATED SCHEMA TO GET THE ADDRESSES OF THE FRAGMENTS AND THEIR LOCAL NAMES

FINISH

FIGURE 9.8 FLOW CHART FOR TRANSACTION DECOMPOSER
START

COLLECT INFORMATION OR ACKNOWLEDGEMENT PIECES FROM ITS LOCAL QUEUE

CHECK THE TRANSACTION DECOMPOSING INDICATIONS

ASSEMBLE THE INFORMATION OR ACKNOWLEDGEMENT PIECES ACCORDING TO TRANSACTION DECOMPOSING INDICATIONS

IS THE ASSEMBLED INFORMATION COMPLETE FOR TRANSACTIONS

Y

DELIVER THAT INTO THE QUEUE FOR MESSAGE ROUTER

NOTIFY THE SYSTEM COORDINATOR

FINISH

CHECK THE DATA DUPLICATION INDICATIONS

IS THE INCOMPLETE ACKNOWLEDGEMENT DUE TO DATA DUPLICATION

Y

FETCH THE RIGHT TRANSACTION

ATTACH TO IT RIGHT ADDRESSES FOR OTHER COPIES

PUT THAT INTO QUEUE FOR MESSAGE ROUTER

N

FIGURE 9.9 FLOW CHART FOR INFORMATION ASSEMBLER
A four schema concept (see figure 9.10) is adopted as the backbone or the reference infrastructure of the prototype information administration system. These schema are:

1. global external schema,
2. global integrated schema,
3. global fragmented schema,
4. global allocated schema.

The global external schema define the information compositions from the viewpoint of each user's interest and how those views are mapped onto the global integrated schema. The global integrated schema define the global structures of the non-repetitive and unified information tables. The global fragmented schema define how the global integrated schema is fragmented and the contents of associated fragmented information tables. The global allocated schema define which physical database those fragmented information tables are allocated to, which fragmented tables are duplicated and where, and their local names of the global fragmented tables and information items. Through the schema manager, the four schema can be retrieved and updated, to enable a high level of configurability (see figure 9.11).

The emerging three schema concept was adopted by US NBS [KRI87] and US Air Force [AF87], which is:

1. global external schema,
2. global conceptual schema,
3. global fragmented schema [KRI87] or global internal schema [AF87].

The global fragmented or internal schema combines what is separately specified in the global fragmented schema and the global allocated schema of the four schema concept proposed by the author. The global fragmented or internal
FIGURE 9.10 FOUR SCHEMA CONCEPT
FIGURE 9.11 FLOW CHART FOR SCHEMA MANAGER
schema specifies what type of information items is stored at each computer node, thus being node-oriented. However, the four schema concept is information-oriented. That is to say, in the three schema approach, questions such as "how many copies of information X are stored in the whole system?" cannot be answered before the whole global fragmented or internal schema is searched, whereas in the four schema approach the global allocated schema specifies clearly the number of copies of all information items and where they are stored. The speed to gain knowledge concerning the number of information copies and where they are stored affects greatly the processing speed of transactions. Therefore, the four schema approach offers opportunities for more efficient processing of transactions than the emerging three schema approach by allowing more straightforward control over information copies.

For the reason of time limitation, the global external schema was not implemented during the time frame of the project. This decision was also based on the understanding that the global external schema is less important than the other three schema, without which, users can see the global integrated schema which encompasses their external schema.

9.4.8 Local DBMS Front End

The role of local DBMS front end is simply to fetch the simple transactions and to call the local DBMS through its SQL interface, thereby processing the transactions. After information or acknowledgements are fed back, the local DBMS front end notifies the system coordinator that information is ready for assembly. (see figure 9.12)
COLLECT A TRANSACTION

CALL DBMS WITH SQL INTERFACE AND PASS THE TRANSACTION TO THE DBMS

WAIT FOR THE INFORMATION OR ACKNOWLEDGEMENT

PUT THAT IN THE QUEUE FOR MESSAGE ROUTER

ANY MORE TRANSACTIONS?

NOTIFY THE SYSTEM COORDINATER

FIGURE 9.12 FLOW CHART FOR LOCAL DBMS FRONT END
9.4.9 SQL Translator

Local retrievals and updates in SQL are a natural way of accessing INGRES since INGRES is supplied with both QUEL and SQL interface facilities. However, the need to retrieve and update information in SQL for the MICROSS DBMS (with its ad hoc language) necessitates a SQL translator to map between SQL commands and ad hoc commands. It is necessary to include an SQL translator for all DBMS which do not support such an SQL interface. The architecture of the SQL translator (see figure 9.13) was chosen so that the SQL translator first carries out SQL parsing, syntax checking and schema validation through interacting with the local schema manager and subsequently translates SQL commands into local DBMS commands.

9.4.10 Embedded SQL Precompiler

From the application programmer's point of view, the access of information can be conveniently realized by embedding SQL statements in application programs with each statement preceded by the symbol "###SQL". The embedded SQL precompiler compiles embedded SQL statements into host high level language statements. The mechanisms for the delivery of transactions to the information administration system and the fetching of information and acknowledgements from the system are entirely transparent to application programmers. Under the UNIX operating system, this can be achieved by employing the "pipe mechanism". Application processes communicate with the message router through "request pipes" and "reply pipes". While "request pipes" buffer transactions, "reply pipes" store information and acknowledgements for subsequent fetching by application processes (see figure 9.14).
FIGURE 9.13 ARCHITECTURE OF SQL TRANSLATOR
SEARCH FOR AN EMBEDDED SQL STATEMENT PROCEEDED BY "##SQL"

IS IT FOUND

END OF THE APPLICATION PROGRAM

IS IT A COMPLETE TRANSACTION

REPORT THE INCORRECT SQL TRANSACTION

REPLACE THE EMBEDDED SQL STATEMENTS WITH 'C' STATEMENTS WHICH PUTS THE TRANSACTION INTO A 'REQUEST PIPE' AND WAITS FOR THE RESULT FROM A 'REPLY PIPE'

FIGURE 9.14 FLOW CHART FOR EMBEDDED SQL PRECOMPILED
9.5 Operational Procedures

9.5.1 Schema Specification and Message Header Format

To provide a deeper understanding of the prototype implementation, example operational procedures of the system will be described to complement the functional explanations of system architecture and its mechanisms. The processing of transactions is dependent on the pre-defined schema information. Three of the four schema were implemented in this system for the reason stated in sub-section 9.3.7, viz: the global integrated schema, the global fragmented schema and the global allocated schema. An example specification of the three schema is shown below (*):

Global Integrated Schema

jobs (job-no description quantity customer due-date DSD components drawing-no priority)
wk-centres (wk-code ...)
wip-components (...) wip-ops (...) work-to-lists (...) job-status (...) scrap (...) (*)

The information types specified in the example global integrated schema are common information passed between and/or among a typical cell and a workshop hierarchical levels in a manufacturing company.
In this particular example, the global integrated schema, involving eight information tables, is specified. The format adopted is as follows: "name of table (column 1, column 2, ...)".

Global Fragmented Schema

define jobs as jobs
define wk-centres as wk-centres
define wip-components as select * from F-WIP-CMP1 where job-no='JOB1' and job-no='JOB2' UNION select * from F-WIP-CMP2 where job-no !='JOB1' and job-no !='JOB2'
define wip-ops as wip-ops
define work-to-lists as work-to-lists
define job-status as job-status
define scrap as scrap

In this chosen example, none of the tables in the global fragmented schema are fragmented except for the table "wip-components" which is fragmented into smaller tables: F-WIP-CMP1 and F-WIP-CMP2.

Global Allocated Schema

store jobs at IBM-PC
store wk-centres at IBM-PC
store F-WIP-CMP1 at IBM-PC SUN
store F-WIP-CMP2 at IBM-PC
store wip-ops at IBM-PC
store job-status at IBM-PC
store scrap at SUN
<table>
<thead>
<tr>
<th>Global name</th>
<th>Local name</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobs ( ... )</td>
<td>jobs ( ... )</td>
</tr>
<tr>
<td>wk-centres ( ... )</td>
<td>wk-centres ( ... )</td>
</tr>
<tr>
<td>F-WIP-CMP1 ( ... )</td>
<td>wip-components ( ... )</td>
</tr>
<tr>
<td>F-WIP-CMP2 ( ... )</td>
<td>wip-components ( ... )</td>
</tr>
<tr>
<td>wip-ops ( ... )</td>
<td>wip-ops ( ... )</td>
</tr>
<tr>
<td>work-to-lists ( ... )</td>
<td>work-to-lists ( ... )</td>
</tr>
<tr>
<td>job-status ( ... )</td>
<td>job-status ( ... )</td>
</tr>
<tr>
<td>scrap ( ... )</td>
<td>scrap ( ... )</td>
</tr>
</tbody>
</table>

Thus, the global allocated schema specifies where each fragment is stored using the format "store 'name of fragment' at 'node address' ", and the mapping of global and local names. The mapping of global and local names specifies the correspondence between the global names and local names both for all fragmented tables and for all the names of columns in each fragmented table. In this example, the local name for "F-WIP-CMP1" or "F-WIP-CMP2" is "wip-components", and all the local names of columns of tables are the same as corresponding global names.

Messages are passed between modules both within a computer node and across many computer nodes. A message starts with the message header for which the format chosen is as follows: "beginning of message indicator, message type, originating node address, originating user name". The format chosen for "beginning of message indicator" was "***" and was used not only to indicate the beginning of a message but also to separate messages whenever they are in the same queue. "Message type" is to ensure the correct routing of transactions
to the schema manager, transaction decomposer, information assembler or to users. Three options were chosen here viz:

(1) "SQLDML" corresponding to the SQL information manipulation language for information accesses,

(2) "SQLDDL" corresponding to the SQL information definition language for system configurability, and

(3) "result" corresponding to replying information or acknowledgement.

SQLDML transactions are routed to the transaction decomposer while SQLDDL transactions are routed to the schema manager. The originating node address and originating user address are jointly used to uniquely identify a user. Incomplete reply messages are delivered to the information assembler and complete reply messages are delivered to users. Message headers are not of concern to users since they are automatically added to and stripped off from users’ transactions by the system and used by the system for the above purposes.

9.5.2 Example Compound Retrievals

Local and remote information retrievals and updates are special cases of compound requests. Thus, the general procedure for processing example compound requests will be explained through an example as follows:

Transaction 1. *** SQLDML IBM-PC Terminal select scrap.*
       work-to-lists.* from work-to-lists, scrap where work-to-lists.wk-code='101' and
       scrap.drawing-no='1'.

The message header "*** SQLDML IBM-PC Terminal" means that this message is a SQLDML transaction, which in the prototype implementation
originates from a terminal user at the IBM-PC. This transaction is first passed to
the message router by the terminal server and then moved to the transaction
decomposer by the message router at the IBM-PC.

In general, the transaction decomposer takes the following steps to decompose a
transaction (see also figure 9.8 and the schema specification in sub-section
9.4.1):

Step 1. validate the information tables according to the global integrated
schema. If they are valid, go to step 2. Otherwise, output an error
message and exit,

Step 2. check how the information tables are fragmented by looking up the
global fragmented schema and hold the names of fragmented tables. If
the search conditions are valid, go to step 3, otherwise, output an error
message and exit,

Step 3. obtain the addresses of the fragmented tables and their local names from
the global allocated schema,

Step 4. decompose the transaction according to the above fragmentation and
allocation information.

The results of these four steps for the example transaction are:

Step 1. Both information table "scrap" and "work-to-lists" are valid.
Step 2. Both information table "scrap" and "work-to-lists" are stored as single
fragments.
Step 3. The address for "work-to-lists" is "IBM-PC" and the address for "scrap"
is "SUN". Their local names are the same as their global names.
Step 4. The decomposer decomposes this transaction into two simple
transactions:
An explanation of how the transaction is decomposed is given as follows: "*** SQLDML IBM-PC Terminal SUN IBM-PC", which must be stored for the information assembler to utilize. The simple transaction corresponding to "work-to-lists" is put into a local queue for processing by the local DBMS front end module in the IBM PC while the transaction corresponding to "scrap" is added into the queue for the communication manager at the IBM PC for shipment to the specified remote node, i.e. the SUN workstation. Once activated by the system coordinator, the information assembler at the IBM PC utilizes the transaction decomposing indication to attempt to assemble the incoming information fragments from the message router and the local DBMS front end. If information fragments are complete for the transaction, they will be assembled together and a message header "*** SQLDML result IBM-PC Terminal" will be attached and delivered to the message router. In the mean time the system coordinator is notified. Then, the message router moves the information to the buffer of the terminal server at the IBM PC. When the terminal server at the IBM PC is active, information will be displayed on the screen for the requesting user.

Let us look at another example transaction which originates from the SUN workstation terminal.

Transaction 2 : *** SQLDML SUN Terminal select wip-components.*
The procedure required to process this transaction (see also figure 9.8 and schema specification in the sub-section 9.4.1) is similar to that for the previous transaction. The table "wip-components" is fragmented into F-WIP-CMPl and F-WIP-CMP2 as indicated in the global fragmented schema. This causes the transaction decomposer to compare the condition of "job-no='JOB1'" for "wip-components" specified in the transaction, with the condition specified for each fragment of "wip-components" in the global fragmented schema. Here, F-WIP-CMPl is the right fragment to trace on for "JOB1" as specified in the global fragmented schema (see sub-section 9.4.1). The global allocated schema shows that F-WIP-CMPl is duplicated on both the IBM-PC and the SUN workstation. Since the transaction originates from the SUN workstation terminal, the SUN workstation was chosen in this case to process F-WIP-CMPl by the transaction decomposer. The local name for F-WIP-CMPl is "wip-components". Therefore, the decomposed simple transactions and transaction decomposing indication are:

Simple transaction 1 : *** SQLDML SUN Terminal select * from wip-components where job-no='JOB1'.

Simple transaction 2 : *** SQLDML SUN Terminal select * from wk-centres.

Transaction decomposing indication : *** SQLDML SUN Terminal SUN IBM-PC.
A description of the remainder of the processing is omitted since this corresponds to that for the previous example.

9.5.3 An Example Embedded SQL Program

An extract of an example embedded SQL program will be presented to illustrate how information can be accessed by application programs. An embedded SQL program (see the example below) has a compound SQL transaction which begins with "Begin transaction" and ends with "End Transaction". This is a very straightforward way for application programmers to access information.

```c
main(argc, argv)
int argc;
char *argv[];
{
    FILE *fp1;
    int i, n, fd1, fd2, fd3, fd4, flag1, flag2, ffd1;
    i = 0;
    while (i)
    {
        ***SQL begin transaction
        ***SQL select wip-ops.*, scrap.* from wip-ops, scrap
        ***SQL where wip-ops.job-no='JOB2' and wip-ops.net-ref='20-100'
        ***SQL end transaction
        system("cat prgm13o");
        i = i + 1;
    }
}
```
What an application programmer needs to do next is to precompile this program into a host language program by calling the embedded SQL precompiler. Once the host language program is compiled into an executable program, it is ready to access information from the heterogeneous distributed factory information administration system.

9.5.4 Example Simple Updates

If, for example, a user issues a transaction for updating 'scrap', i.e. :

"*** SQLDML IBM-PC Terminal update scrap set failed=failed+10
   where drawing-no='101' ".

the transaction decomposer follows a similar general procedure as that for retrievals to process the update transaction (see also figure 9.8 and the schema specification in the sub-section 9.4.1):

step 1. validate the table name by checking the global integrated schema. if they are valid tables, go to step 2, otherwise, report an error message and exit;

step 2. determine the state of fragmentation;

step 3. obtain the addresses of the fragments and their local names;

step 4. carry out the decomposition.

The results of those four steps for the example transaction are :

step 1. table 'scrap' is a valid table;
step 2. table 'scrap' is stored as a single fragment;
step 3. the address for 'scrap' is 'SUN' and its local name is 'scrap';
step 4. the decomposed transaction is the same as the original one:

```sql
*** SQLDML IBM-PC Terminal update scrap set failed=failed+10 where drawing-no='101'.
```

Transaction decomposing indication: *** SQLDML IBM-PC Terminal SUN.

The header "*** SQLDML IBM-PC Terminal" is embedded within the transaction until it is stripped off and held by the local DBMS front end module at the SUN workstation. After the local DBMS front end module receives an acknowledgement, it replaces 'SQLDML' with 'result' in the header, attaches the revised header to the acknowledgement message, notifies the system coordinator and places the acknowledgement message in the queue for message router at the SUN workstation for subsequent distribution to the user at the IBM PC. When this occurs, the information assembler at IBM-PC reports the receipt of an acknowledgement to the user.

9.5.5 Example Updates with Information Duplication

From considering the global allocated schema (see sub-section 4.2.1), we can see that the information fragment 'F-WIP-CMP1' is duplicated at both nodes of the prototype system. Information copies are in danger of being incorrect and inconsistent if some of the copies are not updated or if synchronized updating is not maintained. For instance, suppose that two separate users want to update the number of components completed for 'F-WIP-CMP1' from terminals at both the
IBM-PC and the SUN workstation. If updates are only made locally, the two copies will be certainly in an incorrect and inconsistent state, giving rise to errors. Even if both updates are made on all the copies, the problem of information inconsistency and errors can still arise in some cases.

For certain operations like "addition" and "subtraction", information errors and inconsistency are only temporary as long as updates are propagated to all the copies, regardless of the order of the updates. For example, if the quantity of components to be made is 100, and transaction A from the SUN workstation reports that 10 components have been finished and transaction B reports from the IBM-PC that another 13 components are complete, the information update will result in the following:

<table>
<thead>
<tr>
<th>at the SUN workstation</th>
<th>at the IBM-PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>original number of components</td>
<td>original number of components</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>after transaction A</td>
<td>90</td>
</tr>
<tr>
<td>after transaction B</td>
<td>87</td>
</tr>
<tr>
<td>after transaction A+B</td>
<td>77</td>
</tr>
</tbody>
</table>

After the two transactions are completed on both copies, they will be correct and consistent. However, the synchronized updating of all copies is vital for concurrent updates involving operations like "multiplication" and "division". This was discussed in Chapter 8, but for readability and completeness of discussion, another example will be given below.
Suppose that there are a number of copies of employees’ salaries stored at a number of computers. A manager first decides to increase some employee’s salary by $1000, whose current salary is $200000. Assume that this transaction (which will be referred to as transaction A) is entered via computer terminal A and subsequently the manager decides to increase the same employee’s salary by 10% (transaction B). Before transaction A is propagated to all the copies, transaction B is entered via computer B and is processed first at computer B. Asynchronous updating of different copies of the employee’s salary will cause errors and an inconsistency problem as illustrated below:

<table>
<thead>
<tr>
<th>Computer A</th>
<th>Computer B</th>
</tr>
</thead>
<tbody>
<tr>
<td>current salary</td>
<td>$20000</td>
</tr>
<tr>
<td>after transaction A</td>
<td>$21000</td>
</tr>
<tr>
<td>after transaction A+B</td>
<td>$23100</td>
</tr>
<tr>
<td>current salary</td>
<td>$20000</td>
</tr>
<tr>
<td>after transaction B</td>
<td>$22000</td>
</tr>
<tr>
<td>after transaction A+B</td>
<td>$23000</td>
</tr>
</tbody>
</table>

Obviously, the final value at computer B is not correct because updating required by the two transactions is in the wrong order and this error will be permanent since both transactions are finished.

A sensible approach to solving such problems here is to designate one copy as a primary copy (see also section 8.5) which will be stored at the designated primary node. In the prototype implementation, the SUN workstation was chosen as the primary node since it had a multi-tasking operating system while the IBM PC AT had only a single tasking operating system. The primary copy is always updated first and updates are propagated to all other copies in a synchronized order. Eventually, all copies will reach an error-free and consistent state.

-215-
To illustrate the use of this approach further, consider the following example transaction for updating information with duplication:

```sql
*** SQLDML IBM-PC Terminal update wip-components set quantity =quantity-10 where job-no='JOB1' and net-ref='20-100'.
```

The execution results of the four steps specified in sub-section 4.2.4 are:

step 1. the information table "wip-components" is a valid table,
step 2. the information table "wip-components" is fragmented, and F-WIP-CMP1 is the right fragment for "JOB1",
step 3. F-WIP-CMP1 is duplicated and its local name is "wip-components",
step 4. the decomposed transaction is the same as the original one:

```sql
*** SQLDML IBM-PC Terminal update wip-components set quantity=quantity-10 where job-no='JOB1' and net-ref='20-100'.
```

The transaction described will be distributed to the SUN workstation to update the primary copy. Since there are two copies of the information, the transaction decomposing indication is:

'*** SQLDML IBM-PC Terminal SUN IBM-PC'.

A copy of the transaction is stored for later propagation to the other information copy stored at IBM-PC. The responsibility for propagation is assumed by the information assembler at the primary node. The information assembler at the IBM PC checks whether updates are complete on all copies and issues the
acknowledgement to the user at the IBM PC when all the copies are updated.

9.6 System Configuration

9.6.1 System Configuration for Introduction of New Information Tables

The prototype factory information administration system is highly configurable, resulting from the use of the configurable four schema concept. A further example will be considered here to illustrate this fact.

Suppose that it becomes necessary to store payroll information in the INGRES DBMS as a new type of information or as part of reorganisation aimed at increasing the efficiency with which information can be accessed. For the SUN workstation, the steps below can be followed to accomplish this goal.

step 1. create a table called 'payroll' in INGRES using SQL(*):

```sql
create table payroll ( name c20, salary integer );
```

step 2. create a table called 'payroll' using SQL for the global integrated schema via the schema manager :

(*) The specifications of information tables are the results of EER data modelling. Recall that as described in Chapter 7, the products of EER data modelling are entity relations and relationship relations which are actually information tables. These can be used naturally by an information administrator to define the global integrated schema. Thus, the link between EER data modelling process and schema definition process can be smoothly achieved.
create table payroll ( name c20, salary integer );

step 3. define the fragmentation of the table 'payroll' for the global fragmented schema:

define payroll as payroll;

step 4. define the allocation of the table payroll and carry out the mapping to the local name:

store payroll at SUN

mapping payroll onto payroll.

For the IBM PC, step 1 is not needed. After the configuration procedure has been finished, the table 'payroll' is accessible from any node integrated by the prototype factory information administration system. Terminal users and application programs from any node in the system can insert, update, delete and retrieve "payroll" information.

Deletion of the table 'payroll' is achieved simply by the following configuration steps:

step 1. delete the table 'payroll' in INGRES in SQL:

drop payroll.

step 2. delete 'payroll' in the global integrated schema:

l-drop payroll,

step 3. obtain the fragment names before deleting 'payroll' in the global fragmented schema, then delete "payroll" in the global fragmented schema:

f-drop payroll,

step 4. delete 'payroll' in the global allocated schema:

a-drop payroll.
Then, all the information concerning 'payroll' is destroyed. Step 1 is not needed for the IBM PC.

9.6.2 System Configuration for Transparency under Changes

The four schema mechanism is the key to a number of levels of transparency: allocation transparency, duplication transparency, fragmentation transparency and composition transparency. All of these properties are essential for releasing information users' burdens to know:

1. where all information fragments are stored,
2. whether the information is duplicated,
3. whether the information is stored in multiple tables or in just one table, and
4. what the global integrated information tables are composed of.

What a user needs to know is the structure of those information tables of his/her interest as defined in the global external schema. Moreover, the configuration of the four schema approach ensures that all those transparency capabilities are maintained when information is moved from one node to another, when further information duplications are required, when information tables are refragmented or when the global integrated tables are restructured for reasons such as higher efficiency and reliability. With the four schema approach, what the information administrator needs to do for coping with changes is simply to reconfigure the four schema. Without the four schema approach, it would not be possible to accept changes unless major and costly software modification is carried out.

9.7 System Implementation Mode

The availability of a multi-tasking operating system (i.e. enabling concurrent
execution of a number of processes) provides a convenient tool for implementing the prototype information administration system. Terminal users can access information by interacting with an independent process of terminal server. The communication manager can be started as another independent process, thereby polling the network continuously. In the prototype implementation, the whole transaction processing system runs as the third process (see figure 9.3). This "three process" approach may not be the most efficient for servicing a number of users' requests. It is also doubtful whether the practice of creating a process for every transaction is an efficient one due to the time wasted in creating new processes especially with frequent process switching which the operating system must perform. The optimal set of information administration system processes to service a number of users' requests depends on the nature of the operating system and the frequency and type of users' requests.

An extension to the prototype implementation could be the inclusion of a process creation manager which can assume responsibility for process creation and deletion according to the demands and the nature of operating systems, thereby increasing the efficiency of the information services provided. However, potential for achieving higher efficiency through multiple service processes will introduce complexity. For example, concurrency control mechanism which is necessary for ensuring validity of information stored in databases under the circumstance of multiple system processes must be implemented. A more efficient concurrency control mechanism has been proposed in Chapter 8.

For a single tasking operating system like MS-DOS, only one process can run at a time. This restriction makes it necessary to allocate processor time to a number of concurrent users according to some schedule (see figure 9.4(2)).
this case, the system can only poll the network for remote requests for a limited period of time, then spend some time with a terminal user and process transactions, if there is any, on some cyclic basis.

9.8 Summary

This prototype factory information administration system, which is built upon typical heterogeneous computer hardware, operating systems and DBMSs, possesses the generic functions required to support information accesses from both computer terminals and within application programs. Those information access types include concurrent local/remote compound information retrievals in SQL and concurrent local/remote simple updates in SQL with or without information duplication. Furthermore, a generic information distribution and duplication environment is provided. System configurability can be achieved through the use of a configurable four schema mechanism which ensures that four levels of transparency are provided to assist significantly the users and system builders, i.e. allocation, duplication, fragmentation and composition transparency. The first three types of transparency have been supported in the prototype system while composition transparency can be potentially supported through the subsequent implementation of the global external schema. The transparency capabilities can be provided even when there will be changes like introducing new information tables and reorganizing information storage. The experience in designing and building this prototype system provides a practical basis for the evolution of complete, full scale and total integrated information administration systems for integrated factories of the future.
CHAPTER 10
CONCLUSIONS AND RECOMMENDATIONS
FOR FUTURE WORK

10.1 Research Findings and Conclusions

10.1.1 Introduction

In this section, conclusions are drawn from the study findings and the major contributions to knowledge are highlighted and classified. These contributions complement worldwide CIM systems integration research in the understanding of concepts in the following aspects:

(1) Systems Integration Architecture
* a perception of current stand-alone MEs,
* an architectural specification for systems integration,
* an assessment of MAP/TOP standards.

(2) Design and Prototype Implementation of a FIAS
* emphasis of the importance of database management for systems integration,
* an architectural design and modular specification of a Factory Information Administration System (FIAS),
* the suggested use of SQL (Structured Query Language) as a standard manufacturing database access language and EER (Extended Entity Relationship) data model as a standard manufacturing data modelling method,
* the introduction of a novel concurrency control mechanism for FIASs
based on distributed databases, (this is less important than other contributions)

* the proposal of a four schema information access architecture that

has advantages when compared with the recently favoured use of three
schema architecture,
* the implementation of a prototype heterogeneous FIAS which

provides configurability to support information access transparency even under
major changes.

The above contributions will be considered in greater detail in the following
sub-sections.

10.1.2 Systems Integration Architecture

The analyses of the composition of contemporary or stand-alone MEs and their
interactions have been used to support the formalisation of CIM systems
integration architecture as summarised below.

The term "ME", which was defined in Chapter 2, is used to refer to any
manufacturing sub-system which contributes to product realisation in some way.
Examples of MEs are a CAD system, a financial accounting system, a MRP
system, a scheduling system, a robot and a NC machine. The analysis of
contemporary or stand-alone ME has led to the generic concept concerning the
composition of MEs. For CIM systems integration it is concluded that a
contemporary or stand-alone ME can be perceived as being composed of one or
more information processing sub-systems (which makes decisions and/or deals
with real world state changes) and an information repository which with
contemporary MEs is closely coupled to the information processing
sub-system(s). An information processing sub-system encompasses both "the pure
decision-making sub-system" and "the decision-making sub-system for enabling real world state changes" which can be covered under the generic term "application program". Information exchange is a means by which all application programs interact with each other. An enterprise can be perceived as being composed of application programs and information repositories when considering systems integration. In fact, production processes in an enterprise can be viewed as consisting of discrete and concurrent changes of digital information and the real world states. The problem of integrating contemporary or stand-alone MEs is really a problem of integrating activities of a variety of application programs and information repositories.

In manually integrated manufacturing systems, the interactions between contemporary or stand-alone MEs are accomplished by operators. The role of operators is:

(1) to collect from interrelated application programs within contemporary or stand-alone MEs the relevant information required for making new decisions or controlling machines and to update (insert, modify and delete) the information collected in an information repository coupled with the particular decision-making or control functions (i.e. application programs) to make this information available for decision making or control,

(2) to execute design, management decision-making, and control functions (i.e. application programs) in the right order.

Therefore, if operators are to be replaced by computerized systems integration tools, corresponding to the functions performed by operators, the collective functional requirements of these integration tools are:

(1) to ensure reliable and efficient enterprise-wide digital data transfer among computer nodes,
(2) to reliably and efficiently manage the storage of information generated by distributed application programs and the supply of information required by other distributed application programs,
(3) to automatically activate and coordinate distributed application programs. Requirement (1) is fundamental for meeting requirements (2) and (3) because it specifies the basic need for data communication links between computer nodes.

Corresponding to the three functional requirements specified above, three systems have been proposed to comprise systems integration architecture, viz:
(1) Digital Data Transfer System (DDTS),
(2) Information Administration System (IAS),
(3) Application Administration System (AAS).

The DDTS is to ensure reliable and efficient data communication among networked computer nodes. The IAS is to manage information storage, retrieval and update efficiently and reliably, providing automatic information services for all application programs and operators whenever and wherever necessary. The AAS is to automatically activate and coordinate application programs according to predefined precedence constraints. The relationships among the DDTS, IAS, AAS, application programs and operators, and CIM system programmers are:
(1) both the IAS and AAS utilize data communication services of DDTS,
(2) the IAS can directly provide application programs and operators with information services rather than via the AAS,
(3) the AAS serves CIM system programmers with precedence definitions of application programs and run-time activation and coordination of those application programs, and
(4) there may not necessarily be direct interactions required between the IAS
and the AAS.

The formalisation of the systems integration architecture presented above was achieved jointly with other members of the L.U.T. Systems Integration Group, which was based on the "three architecture" approach proposed by Weston, et.al.. This formalisation provided support, substantiation and enhancement to the "three architecture" approach by analysing, from the viewpoint of integration, the composition of contemporary or stand-alone MEs and their interactions in manually integrated systems. Based on the analyses, more precise architectural specification was also attempted as presented above.

By dividing the complex CIM systems integration problem into the design and implementation of three systems as stated above, major advantages accrue as follows:

(1) CIM systems integration becomes conceptually clearer and more manageable.

(2) The CIM systems integration problem is decomposed so that expertise can be brought to bear on each area, viz: the knowledge from the manufacturing community on the AAS and the formulation of the IAS and DDTS by the database (which will be explained later) and digital communication communities respectively.

(3) The decomposition allows much greater flexibility because different mechanisms can be developed to realise each of the three systems without affecting the other two systems.

During this study, the author has also sought to assess the role of MAP/TOP for CIM systems integration. In the latest version of MAP/TOP (version 3.0), the services provided by the application layer of MAP are MMS (Manufacturing
Messaging Services) and FTAM (File Transfer, Access and Management) while FTAM and MHS (Message Handling Service) are the services within the application layer of TOP. FTAM can provide suitable services to support a network-wide digital data transfer system. Furthermore, it has been shown that to some extent, MMS can accomplish some of the administrative functions required by the application administration system. MHS has been devised to provide electronic mail services. Arguably, the biggest gap yet to be bridged is the lack of a suitable information administration system.

10.1.3 Design and Prototype Implementation of a FIAS

The lack of a suitable information administration system will severely limit the evolution of flexibly integrated manufacturing systems. Therefore, much of this project study has been targeted on this area. This work has complemented the work of other members of the L.U.T. Systems Integration Group, where their work has largely been aimed at deriving novel forms of the AAS based on a MAP implementation of the DOTS. Previous work in the database management has been aimed primarily at solving information administration problems. The author strongly emphasizes the importance of database management technology for establishing systems integration. Sadly, there is often a lack of awareness of the importance of this point within the CIM systems integration community, this being reflected in the relative lack of literature relating to the exploitation of this technology for establishing manufacturing systems integration. The elaboration of this point is one of the most important contributions of this thesis, where the author's implementation studies provide a validation of the arguments presented. There is a fast growing need to marry the technologies and experts associated with the database and CIM systems integration community respectively.
The pre-requisite for designing and implementing an efficient, reliable and cost-effective FIAS is the de-coupling of information repositories from contemporary or stand-alone MEs. Furthermore, proper DBMSs are suggested to be utilised.

This thesis has also considered the relative importance of centralized database approach and distributed database approach in building FIASs. The distributed database approach has many advantages, when compared with a centralized database approach, resulting from:

(1) storage limitation of a single computer,
(2) existing distributed databases,
(3) potential reduction in communication costs,
(4) potential quicker responses to users' information requests,
(5) potentially higher reliability and availability.

This thesis has shown that these advantages can become reality through using "decentralization favoured administration", where only the concurrency control mechanism is centralized.

The functional modules of a Factory Information Administratrtion System (FIAS) specified by the author are as follows:

(1) process creation manager,
(2) system coordinator,
(3) terminal server,
(4) embedded SQL precompiler,
(5) message router,
(6) communication manager,
(7) schema manager,
(8) transaction decomposer and processing optimizer,
(9) security controller,
(10) transaction execution manager,
(11) crash recovery organizer,
(12) concurrency controller,
(13) information assembler,
(14) local DBMS front end,
(15) information access statistics monitor,
(16) SQL translator, and
(17) information translator.

The co-existence of DBMSs of various types is inevitable, and an "information standard approach" has been proved exhaustively as being the most productive way of coping with the co-existence of DBMSs of various types. By considering historical developments of database access languages and data modelling methods and making predictions about the trends of the technology for the next decade, SQL (Structured Query Language) was proposed by the author as the standard manufacturing database access language and the EER (Extended Entity-Relationship) data model as the standard manufacturing data modelling method, which could be used possibly beyond the next decade. The principle for the proposals is "commercial porpularity during the intended duration of the standard". These proposals were made in supporting PDES/STEP initiative and hopefully they will be recognised by the PDES/STEP committee. The standardization of manufacturing information formats will largely depend upon the efforts of the PDES/STEP initiative.

The need for information translations has been addressed from three perspectives:
(1) the network communication perspective (i.e. at application layer of networking),

(2) the distributed database management perspective,

(3) the perspective from direct information transfer between CAD/CAM databases.

The three perspectives were from the computer network communication community, the database community and the manufacturing community respectively. The three communities are suggested to combine their efforts to solve the common problem of "disparity in information formats" through information translations. The author further suggests that the most appropriate stage at which this translation should be carried out is the stage seen from distributed database management perspective by providing each DBMS and application program with a standard SQL translator and a standard manufacturing information translator.

In seeking a suitable concurrency control mechanism for multi-user FIASs based on distributed databases, various concurrency control mechanisms were reviewed, evaluated qualitatively and compared. A multiple primary and improved 2PL (Two Phase Locking) concurrency control mechanism was proposed for distributed databases used in creating FIASs. The proposal has been made based on the following findings:

(1) 2PL is the best candidate for efficiency improvement among the ten concurrency control mechanisms reviewed.

(2) 2PL can be improved by breaking down long transaction waiting chain.

(3) Primary copy approach for distributed databases is the most efficient but least reliable, whereas the multiple primary copy approach provides the balance of efficiency and reliability.
A four schema architecture for information access was also proposed for FIASs, where the four schema are:

1. Global external schema,
2. Global integrated schema,
3. Global fragmented schema,

These schema can be compared with the emerging conventional three schema approach, where the three schema are:

1. Global external schema,
2. Global conceptual schema,
3. Global internal or fragmented schema.

The three schema approach combines the global fragmented schema and the global allocated schema within the global internal or fragmented schema. A four schema architecture is conceptually clearer than a three schema architecture and offers opportunities for more efficient transaction processing by allowing more straightforward control over information copies.

A prototype heterogeneous FIAS has been implemented, which integrates a relational DBMS having an SQL interface (called INGRES run on a SUN workstation) and a specific DBMS having no SQL interface (called MICROSS run on an IBM PC AT). This FIAS facilitates concurrent compound information retrievals in SQL and concurrent simple information updates in SQL. The configurable schema provide the system with fragmentation, duplication, allocation transparency even under major changes. The configurability of FIASs is essential for dealing with changes in hardware and software without affecting existing systems, being important to the success of Factory 2000. The experience gained provided a practical basis for constructing a
full scale information administration systems for the integration of factories of the future

10.2 Recommendations for Future Work

The author, along with other members of the LUT systems integration group, believes that segmentation of integration infrastructure into network digital data transfer system, information administration system and application administration system is essential. In the author's opinion, the creation of an efficient, reliable and cost-effective FIAS is the major task of CIM systems integration. Emerging distributed database technology must be harnessed in creating the information administration system. Since the basic technology associated with distributed database management is available, although not yet perfected, the major problems left relate to implementation issues. Here recommendations for future research study will be made both for the world-wide CIM systems integration community and specifically for the systems integration project work in Dept. of Manufacturing Engineering at Loughborough University of Technology (LUT):

For the world-wide CIM systems integration community:

(1) It is strongly recommended that activities of the CIM systems integration community be integrated with those of the database community. CIM systems integration is one of the most important application areas for the exploitation and development of distributed database technology.

For the LUT systems integration group, the author's work can be extended as follows:

(1) The integration environment around the LUT flexible assembly cell is
recommended to be extended to incorporate the various design, factory, shop and cell computer systems and machine gateway systems, which are linked by a MAP network. Multiple DBMSs to be used are suggested to be either purchased or created in-house, involving various access languages, operating systems and computer hardware. This kind of environment would be sufficiently generic to evolving world-leading methodologies in factory information administration.

(2) For the creation of a FIAS, it is recommended that the innovations reported in the thesis should be enhanced and fully demonstrated through implementing:

(2.1) "decentralization favoured administration" with only the concurrency control mechanism being centralized,

(2.2) the global external schema of the four schema architecture,

(2.3) SQL translators for all non-SQL DBMSs,

(2.4) multiple primary and improved 2PL (two phase locking) concurrency control mechanism,

(2.5) process creation managers for multi-user computer systems,

(2.6) more comprehensive transaction decomposer and processing optimizers,

(2.7) crash recovery organizers.

The suggested arrangements could lead to commercial products providing heterarchical information administration in a variety of heterogeneous manufacturing organisations.
REFERENCES


[BEN85] Bennett, R.G., "What are companies spending on CIM and How are they justifying these expenditures", A program guide for CIM implementation, Published by American Technical Publishers Ltd, 1985, pp 55-57.


[BER79] Bernstein, P.A. and Goodman, N., "Approaches to concurrency control


[COD70] Codd, E.F. "A Relational Model of Data for Large Shared Data
Banks", CACM 13, No. 6, June, 1970.


[CRU85b] Cruz, F.D., Fischer, H., and Damens, J., UNIX KERMIT Manual,


[DAT86] Date, C.J., An Introduction to Database Systems, Published by Addison-Wesley, 1986, pp413-429.


[DOR83] Dorf, R.C., Robotics and Automated Manufacturing, Published by Reston, 1983.


[EVE86] Everest, G., Database Management Objectives, System Functions and
Administration, Published by McGraw-Hill, 1986.


[HAL85] Halsall, F., Introduction to Data Communications and Computer Networks, Published by Addison-Wesley publishing company Inc., 1985, pp202 - 244.


[HAR79] Harrington, J., "Can I justify the cost?", Computer Integrated Manufacturing, Malabar, Florida, Rebert E., Published by Krieger


[HUB85] Hubbard, S.W., CAD/CAM Applications for Business, Published by Oryx, 1985.


[ING85] Ingersoll Engineers, Integrated Manufacturing, 1985, Published by IFS(Publications) Ltd., UK.


[LAW84] Lawrence, K.D., Production Planning and Scheduling: Mathematical Programming Applications, Published by Industrial Engineering and


[MAI88] Mairs, C., "Interworking with LU Type 6.2", Networks for the 1990s, Published by Online, 1988, pp122-138.


[MAY81] Mayne, A., Database Management Systems-à Technical Review,


[O'GR86]O'Grady, P.J., Controlling Automated Manufacturing Systems,
Published by Kogan Page Ltd, 1986.


[PUS83] Pusztai, J., Computer Numerical Control, Published by Reston, 1983.


with Case Studies, 1986, Published by Prentice-Hall International, UK, Ltd.

[RAT85] Rathmill, K., Robotic Assembly (industrial trends in manufacturing technology), Published by IFS, 1985.


[TAY76] Taylor, R.W., Frank, R.L., "CODASYL Database Management
Systems", ACM Comp. Surv., Vol 8, No.1, Mar 1976, pp76-103.


[WOR85] Works, M.T., "Cost justification and new technology addressing managements' "No!" to the funding of CIM", A Program Guide for CIM Implementation, 1985, Published by American Technical

APPENDIX A

SELECTED FUNCTIONAL DESCRIPTION OF THE KERMIT FILE TRANSFER

The background of KERMIT file transfer has already been presented in Chapter 2 and a detailed technical description can be found in the documentation authored by Cruz [CRU85a] [CRU85b]. Here, the functional properties of KERMIT-MS and C-KERMIT will be outlined, and subsequently the available procedures for transferring a file will be detailed. The KERMIT file transfer mechanism was utilized in the implementation of the prototype Factory Information Administration System (see Chapter 9).

<table>
<thead>
<tr>
<th>MS-DOS</th>
<th>KERMIT Function List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bye</td>
<td>Log</td>
</tr>
<tr>
<td>Clear</td>
<td>Output text</td>
</tr>
<tr>
<td>CIRinp</td>
<td>Push</td>
</tr>
<tr>
<td>CWD</td>
<td>Receive</td>
</tr>
<tr>
<td>Delete</td>
<td>Run</td>
</tr>
<tr>
<td>Do</td>
<td>Server</td>
</tr>
<tr>
<td>Exit</td>
<td>Show</td>
</tr>
<tr>
<td>Get</td>
<td>Status</td>
</tr>
<tr>
<td>Help</td>
<td>Transmit filespec</td>
</tr>
<tr>
<td>Local</td>
<td>Version</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(IBM-PC Kermit-MS V2.29a)
UNIX KERMIT Function List

( C-Kermit V4.2 )

!     bye     close     connect
cwd   dial    directory  echo
exit  finish  get       help
log   quit    receive   remote
script send    server    set
show  space    statistics  take

After initialization of both MS-DOS KERMIT and UNIX KERMIT (e.g. communication line setting, transmission speed setting, parity bit setting) is complete, two methods can be taken to transfer a file. One method is to set one site in "receive mode" by issuing the command "receive" and to send a file at the other site by issuing the command "send file-name". The other method, which was the mechanism used in the prototype implementation, is to set one site in "server mode" by issuing the command "server" and to send a file at the other site by issuing the command "send file-name" or to obtain a file stored at the site in "server mode" by issuing the command "get file-name". Thus, files, as they are, can be transferred bi-directionally between two computers linked via KERMIT.

References:


APPENDIX B

UTILISED REPORT DETAILS OF MICROSS PP&C PACKAGE

Typical reports, generated by MICROSS PP&C package, were utilized as examples of users' information requirements in the development of a subset of the PP&C Entity Relationship data model in Chapter 7. The following tables are example report details from the MICROSS PP&C package. Those reports were mainly related to scheduling and work_in_progress.

CUSTOMER ORDERS REPORT

<table>
<thead>
<tr>
<th>Job number</th>
<th>Description</th>
<th>Quantity</th>
<th>Customer</th>
<th>Due date</th>
<th>DSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB1</td>
<td>COMPUTER1</td>
<td>100</td>
<td>ICL</td>
<td>02.1.87</td>
<td></td>
</tr>
<tr>
<td>JOB2</td>
<td>COMPUTER2</td>
<td>200</td>
<td>IBM</td>
<td>02.2.87</td>
<td></td>
</tr>
<tr>
<td>JOB3</td>
<td>COMPUTER3</td>
<td>300</td>
<td>KEWILL</td>
<td>02.3.87</td>
<td></td>
</tr>
<tr>
<td>JOB4</td>
<td>VDU</td>
<td>50</td>
<td>JAK</td>
<td>02.1.87</td>
<td></td>
</tr>
<tr>
<td>JOB5</td>
<td>PCB1</td>
<td>100</td>
<td>RUI</td>
<td>02.4.87</td>
<td></td>
</tr>
<tr>
<td>JOB6</td>
<td>COMPUTER_X</td>
<td>200</td>
<td>RUI</td>
<td>10.3.89</td>
<td></td>
</tr>
<tr>
<td>WORK</td>
<td>CENTRE</td>
<td>DESCRIPTION</td>
<td>CAPACITY</td>
<td>UNITS</td>
<td>DEF</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>--------------</td>
<td>----------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>101</td>
<td>AGV1</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>AGV2</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>AGV3</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>ADEPT</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>202</td>
<td>ADEPT2</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>ASEA1</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>ASEA2</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>UNIMATE1</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>TOSHIBA1</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>FRONC1</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>301</td>
<td>ROBOT1</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>302</td>
<td>ROBOT2</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>303</td>
<td>ROBOT3</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>304</td>
<td>ROBOT4</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>305</td>
<td>ROBOT5</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>306</td>
<td>ROBOT6</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>307</td>
<td>ROBOT7</td>
<td>L</td>
<td>1</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>308</td>
<td>ROBOT8</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
OPERATION REPORT

Drawing no. 101 PCB1 Quantity 1

<table>
<thead>
<tr>
<th>OP NO</th>
<th>WORK CENTRE</th>
<th>SETUP</th>
<th>UNIT</th>
<th>TRANSIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>ADEPT1</td>
<td>1.00</td>
<td>0.0300</td>
<td>0.1</td>
</tr>
<tr>
<td>8</td>
<td>ASEA1</td>
<td>0.00</td>
<td>0.0020</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>UNIMATE1</td>
<td>0.03</td>
<td>0.0100</td>
<td>0.0</td>
</tr>
<tr>
<td>12</td>
<td>FRONC1</td>
<td>0.02</td>
<td>0.0100</td>
<td>0.0</td>
</tr>
</tbody>
</table>
WORK TO LIST REPORT

Work Centre CODE ........ 101
Work Centre NAME ........ AGV1
No. of UNITS ............. 1
Performance Factor ....... 100

<table>
<thead>
<tr>
<th>Drawing</th>
<th>Job no &amp; op</th>
<th>TIME</th>
<th>SSD</th>
<th>SFD</th>
<th>Previous</th>
<th>Next</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>job number</td>
<td>Description</td>
<td>Qty</td>
<td>no hrs</td>
<td>W/C</td>
<td>W/C</td>
<td>W/C</td>
</tr>
<tr>
<td>PCB2</td>
<td>JOB3 102 &amp; 255</td>
<td>1</td>
<td>25.6</td>
<td>2.3</td>
<td>3.2</td>
<td>102</td>
<td>3.2</td>
</tr>
<tr>
<td>BOX3</td>
<td>JOB3 107 &amp; 300</td>
<td>1</td>
<td>6.1</td>
<td>3.2</td>
<td>3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOX2</td>
<td>JOB2 106 &amp; 200</td>
<td>1</td>
<td>4.1</td>
<td>4.3</td>
<td>4.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCB3</td>
<td>JOB1 103 &amp; 195</td>
<td>1</td>
<td>19.6</td>
<td>5.1</td>
<td>5.2</td>
<td>102</td>
<td>11.5</td>
</tr>
<tr>
<td>BOX1</td>
<td>JOB1 104 &amp; 200</td>
<td>1</td>
<td>4.1</td>
<td>5.2</td>
<td>5.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCB2</td>
<td>JOB5 205 &amp; R1 1000</td>
<td>1</td>
<td>40.0</td>
<td>5.2</td>
<td>6.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JOB3 102 &amp; 255</td>
<td>5</td>
<td>25.7</td>
<td>8.5</td>
<td>9.3</td>
<td>103</td>
<td>102</td>
</tr>
</tbody>
</table>
### WORK LOAD REPORT

**Work Centre**.............. 101 AGV1
**No. of UNITS**............. 1 PF 100%

<table>
<thead>
<tr>
<th>Week</th>
<th>Available Capacity in hours</th>
<th>Load (in hours) based upon nos</th>
<th>NORMAL</th>
<th>PLANNED</th>
<th>CUMUL</th>
<th>ESD</th>
<th>LSD</th>
<th>cumul</th>
<th>SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>195</td>
<td>177</td>
<td>177</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>51</td>
<td>69</td>
<td>246</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>40</td>
<td>120</td>
<td>40</td>
<td>40</td>
<td>286</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>40</td>
<td>160</td>
<td>30</td>
<td>30</td>
<td>316</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>40</td>
<td>200</td>
<td>0</td>
<td>0</td>
<td>316</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>40</td>
<td>240</td>
<td>0</td>
<td>0</td>
<td>316</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-10</td>
<td>160</td>
<td>160</td>
<td>400</td>
<td>60</td>
<td>60</td>
<td>376</td>
<td>111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-14</td>
<td>152</td>
<td>152</td>
<td>552</td>
<td>0</td>
<td>0</td>
<td>376</td>
<td>51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td>144</td>
<td>144</td>
<td>696</td>
<td>0</td>
<td>0</td>
<td>376</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-22</td>
<td>152</td>
<td>152</td>
<td>848</td>
<td>0</td>
<td>0</td>
<td>376</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| SUB-TOTALS | 376 | 376 | 376 |
| LATER       | 0   | 0   | 0   |

| TOTALS       | 376 | 376 | 376 |

- 255 -
# OVERDUE JOBS REPORT

(Jobs more than 7 days late)

<table>
<thead>
<tr>
<th>No.</th>
<th>REF. No.</th>
<th>DATE</th>
<th>PROB</th>
<th>LATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ww.d.yy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ww.d.yy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ww.d.yy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JOB NET WORK</th>
<th>DWG</th>
<th>DESCRIPT.</th>
<th>QTY</th>
<th>DLY</th>
<th>EST</th>
<th>SCHD</th>
<th>DAYS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No.</th>
<th>REF. No.</th>
<th>DATE</th>
<th>PROB</th>
<th>LATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ww.d.yy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ww.d.yy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ww.d.yy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Job1 | 100-100 | 1 | Computer1 | 200 | 02.1.87 | 04.1.87 | 14.2.87 | 85 |
| Job2 | 100-102 | 2 | Computer2 | 200 | 02.2.87 | 05.3.87 | 13.4.87 | 79 |
| Job3 | 100-103 | 3 | Computer3 | 300 | 02.3.87 | 12.3.87 | 12.3.87 | 70 |
| Job5 | 100-101 | 101 | PCB1 | 100 | 02.4.87 | 02.1.87 | 06.3.87 | 27 |

# JOB STATUS REPORT

<table>
<thead>
<tr>
<th>Job</th>
<th>COMPUTER1</th>
<th>Customer</th>
<th>Batch Quantity</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB1</td>
<td>COMPUTER1</td>
<td>ICL</td>
<td>100</td>
<td>02.1.87</td>
</tr>
</tbody>
</table>

-256-
<table>
<thead>
<tr>
<th>ref number</th>
<th>Description</th>
<th>Qty</th>
<th>ESD</th>
<th>LSD</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>20-100</td>
<td>PCB2</td>
<td>170</td>
<td>1.1</td>
<td>51.3</td>
<td>12.1</td>
<td>13.1.87</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Op no</th>
<th>Work Centre</th>
<th>M/C Time</th>
<th>Trans</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>103 Operation completed</td>
<td></td>
<td></td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>102 AGV2</td>
<td>34.3 hrs</td>
<td>0.1</td>
<td>12.1</td>
<td>12.3</td>
</tr>
<tr>
<td>5</td>
<td>101 AGV1</td>
<td>17.2 hrs</td>
<td>0.2</td>
<td>12.3</td>
<td>12.5</td>
</tr>
<tr>
<td>6</td>
<td>102 AGV2</td>
<td>5.4 hrs</td>
<td>0.1</td>
<td>12.5</td>
<td>12.5</td>
</tr>
<tr>
<td>8</td>
<td>203 ASEA1</td>
<td>0.3 hrs</td>
<td>0.0</td>
<td>12.5</td>
<td>12.5</td>
</tr>
<tr>
<td>10</td>
<td>205 UNIMATE1</td>
<td>1.7 hrs</td>
<td>0.0</td>
<td>12.5</td>
<td>12.5</td>
</tr>
<tr>
<td>11</td>
<td>206 TOSHIBA1</td>
<td>5.1 hrs</td>
<td>0.0</td>
<td>12.5</td>
<td>13.1</td>
</tr>
<tr>
<td>12</td>
<td>207 FRONC1</td>
<td>1.7 hrs</td>
<td>0.0</td>
<td>13.1</td>
<td>13.1</td>
</tr>
<tr>
<td>13</td>
<td>205 UNIMATE1</td>
<td>3.7 hrs</td>
<td>0.0</td>
<td>13.1</td>
<td>13.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ref number</th>
<th>Description</th>
<th>Qty</th>
<th>ESD</th>
<th>LSD</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-100</td>
<td>PCB3</td>
<td>195</td>
<td>1.1</td>
<td>51.3</td>
<td>5.1</td>
<td>13.4.87</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Op no</th>
<th>Work Centre</th>
<th>M/C Time</th>
<th>Trans</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101 AGV1</td>
<td>19.6 hrs</td>
<td>0.1</td>
<td>5.1</td>
<td>5.2</td>
</tr>
<tr>
<td>2</td>
<td>102 AGV2</td>
<td>3.9 hrs</td>
<td>0.1</td>
<td>11.5</td>
<td>12.1</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Qty</td>
<td>M/C Time</td>
<td>Trans</td>
<td>SSD</td>
</tr>
<tr>
<td>---</td>
<td>--------------</td>
<td>------</td>
<td>----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>3</td>
<td>103 AGV3</td>
<td>19.7 hrs</td>
<td>0.2</td>
<td>12.1</td>
<td>12.2</td>
</tr>
<tr>
<td>4</td>
<td>102 AGV2</td>
<td>39.3 hrs</td>
<td>0.1</td>
<td>12.4</td>
<td>13.1</td>
</tr>
<tr>
<td>5</td>
<td>101 AGV1</td>
<td>19.7 hrs</td>
<td>0.2</td>
<td>13.1</td>
<td>13.2</td>
</tr>
<tr>
<td>6</td>
<td>201 ADEPT1</td>
<td>6.9 hrs</td>
<td>0.1</td>
<td>13.2</td>
<td>13.3</td>
</tr>
<tr>
<td>7</td>
<td>202 ADEPT2</td>
<td>4.3 hrs</td>
<td>0.0</td>
<td>13.3</td>
<td>13.3</td>
</tr>
<tr>
<td>9</td>
<td>204 ASEA2</td>
<td>0.2 hrs</td>
<td>0.0</td>
<td>13.3</td>
<td>13.3</td>
</tr>
<tr>
<td>11</td>
<td>206 TOSHIBA1</td>
<td>5.9 hrs</td>
<td>0.0</td>
<td>13.3</td>
<td>13.4</td>
</tr>
<tr>
<td>12</td>
<td>207 FRONC1</td>
<td>2.0 hrs</td>
<td>0.0</td>
<td>13.4</td>
<td>13.4</td>
</tr>
<tr>
<td>13</td>
<td>205 UNIMATE1</td>
<td>4.2 hrs</td>
<td>0.0</td>
<td>13.4</td>
<td>13.4</td>
</tr>
</tbody>
</table>

Network Drawing

<table>
<thead>
<tr>
<th>ref number</th>
<th>Description</th>
<th>Qty</th>
<th>ESD</th>
<th>LSD</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>40-100</td>
<td>BOX1</td>
<td>200</td>
<td>1.1</td>
<td>1.2</td>
<td>5.2</td>
<td>05.2.87</td>
</tr>
</tbody>
</table>

Op no Work Centre M/C Time Trans SSD SFD

|   | 101 AGV1 | 4.1 hrs | 0.0 | 5.2 | 5.2 |

Network Drawing

<table>
<thead>
<tr>
<th>ref number</th>
<th>Description</th>
<th>Qty</th>
<th>ESD</th>
<th>LSD</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>50-100</td>
<td>VDU</td>
<td>200</td>
<td>1.1</td>
<td>1.2</td>
<td>1.2</td>
<td>01.2.87</td>
</tr>
<tr>
<td>Op no</td>
<td>Work Centre</td>
<td>M/C Time</td>
<td>Trans</td>
<td>SSD</td>
<td>SFD</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>----------</td>
<td>-------</td>
<td>-----</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>102 AGV2</td>
<td>4.0 hrs</td>
<td>0.0</td>
<td>1.2</td>
<td>1.2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Drawing</th>
<th>ESD</th>
<th>LSD</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref number</td>
<td>Qty</td>
<td>ww.d</td>
<td>ww.d</td>
<td>ww.d</td>
</tr>
</tbody>
</table>

| 100-101 | 1 COMPUTER1 | 200 | 2.4 | 1.2 | 13.4 | 14.2 | 87 |

<table>
<thead>
<tr>
<th>Op no</th>
<th>Work Centre</th>
<th>M/C Time</th>
<th>Trans</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>301 ROBOT1</td>
<td>40.0 hrs</td>
<td>0.0</td>
<td>13.4</td>
<td>14.1</td>
</tr>
<tr>
<td>3</td>
<td>302 ROBOT2</td>
<td>4.0 hrs</td>
<td>0.0</td>
<td>14.1</td>
<td>14.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Drawing</th>
<th>ESD</th>
<th>LSD</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref number</td>
<td>Qty</td>
<td>ww.d</td>
<td>ww.d</td>
<td>ww.d</td>
</tr>
</tbody>
</table>

| 100-101 | 1 COMPUTER1 | 200 | 2.4 | 1.2 | 13.4 | 14.2 | 87 |

<table>
<thead>
<tr>
<th>Op no</th>
<th>Work Centre</th>
<th>M/C Time</th>
<th>Trans</th>
<th>SSD</th>
<th>SFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>303 ROBOT3</td>
<td>8.0 hrs</td>
<td>0.0</td>
<td>14.1</td>
<td>14.2</td>
</tr>
<tr>
<td>7</td>
<td>304 ROBOT4</td>
<td>4.0 hrs</td>
<td>0.0</td>
<td>14.2</td>
<td>14.2</td>
</tr>
<tr>
<td>9</td>
<td>305 ROBOT5</td>
<td>4.1 hrs</td>
<td>0.0</td>
<td>14.2</td>
<td>14.2</td>
</tr>
</tbody>
</table>
APPENDIX C

AUTHOR’S PUBLICATIONS

Ruj, A., Weston, R.H., Gascoigne, J.D., Hodgson, A. and Sumpter, C.M.,

Weston, R.H., Gascoigne, J.D., Ruj, A., Hodgson, A., Sumpter, C.M. and Coutts, I.,
