A component-based approach to design and construction of change capable manufacturing cell control systems

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 of Loughborough University.

Metadata Record: https://dspace.lboro.ac.uk/2134/11060

Publisher: © R.P. Monfared

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.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
Please note that fines are charged on ALL overdue items.
A Component-Based Approach to Design and Construction of Change Capable Manufacturing Cell Control Systems

By

Radmehr Pourtafreshi Monfared

Doctoral Thesis
Submitted in partial fulfilment of the requirements
for the award of
Doctor of Philosophy
of Loughborough University

January 2000

© by R.P.Monfared - January 2000
Dedicate to my wife

who has devoted her life to our family

and

in memory of Kambiz Zolfaghari
Acknowledgements

I heartily thank God Almighty for his grace that enabled me to complete this work.

I am deeply indebted to my supervisor, Professor Richard H. Weston, for his invaluable and amicable guidance, encouragement and help during completion of this research. Indeed, this research could not be completed without his comprehensive support.

I am also grateful to the Ministry of Culture and Higher Education of Iran for its financial support.

I would like to express my appreciation to my friends and colleagues in MSI Research Institute for their constant support whenever I needed. I sincerely thank current members and those who have left MSI Research Institute, for their friendship, help and for making me feel welcome in this institute.

My especial gratitude should be made to Mrs. Margaret Carden for her kind help.

Credit should be paid to Dr. Marcus Aguiar and Mr. Ian Coutts for their work on SEWOSA and CIMBIOSYS respectively, which have been widely used in this research.

I also would like to thank Dr. Shahin Rahimifard and Dr. Brian Bowen for the constructive discussions that we had during my research study.

Finally, I am most profoundly thankful to my wife and my daughter for their patience, encouragement and love.
Synopsis

Business goals of manufacturing systems are typically in a state of constant change and greater rates of change are predicted in the future. Whereas contemporary approaches to the design and construction of these systems often results in inflexible enterprises that cannot readily be tuned to changing business goals.

This study has specified and prototyped the use of a new model-driven approach to the design and (re)configuration of “change capable” manufacturing cells. Manufacturing cells represent a typical domain of manufacturing systems in which the existence of inflexible links between tasks and resources can result in sub-optimal performance and an inability to cope with change.

The approach is based on a) the use of a semi-generic model of manufacturing cells, that structures and targets the use of CIMOSA modelling constructs (as implemented by the SEWOSA tool) towards producing a requirements specification and conceptual design in the form of a graphical and computer executable model of a particular manufacturing cell, and b) the complementary use of new computer executable modelling constructs and tools, that structure and support the detailed design and runtime operation of a particular cell in the form of an explicit, model-based configuration of cell resources and software components that realise the control processes required in a particular cell.

Part of the semi-generic model comprises descriptions of common tasks found in a given domain of manufacturing cells. That part of the model has been captured and formalised by using CIMOSA modelling constructs. A new development of this modelling structure allows pre-modelled tasks to be selected, detailed and organised and suitable resources and reusable control system components (or building blocks) assigned to groups of tasks. Thereby this new approach to designing and building manufacturing cells can facilitate rapid and effective design and reconfiguration of manufacturing cell control systems. General information requirements found during the modelling and real world application of target cells, have also been formally defined and are met by using a suitable modelling structure and specially developed tools. Furthermore, the research has shown how modelled sets of software component building blocks can be specified and implemented as modular, reusable elements of manufacturing cell control systems.

New modelling structures have been conceived and formalised and examples of their use evaluated under laboratory conditions. The research has also deployed and developed pre-existing enterprise modelling concepts and integration tools, including CIMOSA, STEP, EXPRESS, CIMBIOSYS infrastructure services and component-based software design concepts. This has enable the creation of a prototype tool-set that demonstrates how the concepts can be beneficially applied.

The main contributions made by this research are that:

a) It proposes and develops an approach to the design of manufacturing cell systems that
successfully bridges a previous gap between top-down modelling concepts, methods and tools (that typically support formal modelling of system requirements, tasks and resources) and bottom-up detailed design and build techniques that lead to the operation, control and monitoring of real cells,

b) It provides a modelling and implementation structure that ‘integrates’ the use of a classical enterprise modelling approach (namely CIMOSA), design primarily to support the designers of manufacturing systems, to the emerging component-based design and build concepts, that are becoming popular with software and system vendors.

Keywords:
CIMOSA, Enterprise Modelling, Information Modelling, EXPRESS, Component-Based Design, Software Component, Change-Capable Manufacturing Cells, Design Reconfiguration Methods
Table of Content

Acknowledgements ........................................................................................................................................... 5
Synopsis ............................................................................................................................................................ 6
Table of Content ........................................................................................................................................... 8
List of Figures ............................................................................................................................................... 11
List of Tables ................................................................................................................................................ 13
Glossary of Terms ....................................................................................................................................... 14
Style of Presentation ................................................................................................................................... 15

Chapter 1 : Business Trends and their Impact on the Requirements of Manufacturing Systems ........................................................................................................................................... 16
1.0) Introduction .......................................................................................................................................... 16
1.1) Needs for Change Capability .............................................................................................................. 17
1.2) Need to Deploy a New Approach to System Design and Reconfiguration ......................................... 19
1.2.1) Model of Change ............................................................................................................................ 21
1.3) Common Problems Associated with Current Manufacturing Systems ............................................. 25
1.4) Possible Solution ................................................................................................................................ 26
1.4.1) Modelling of Manufacturing Systems .............................................................................................. 29
1.5) Research Boundaries and Requirements ............................................................................................. 29
Summary ...................................................................................................................................................... 31

Chapter 2 : Literature Review ......................................................................................................................... 32
2.0) Introduction .......................................................................................................................................... 32
2.1) Manufacturing Systems - Definition of Terms .................................................................................... 32
2.1.1) Manufacturing Cells ....................................................................................................................... 33
2.1.2) Cell Controllers ................................................................................................................................ 35
2.2) Enterprise Integration ........................................................................................................................... 37
2.3) Enterprise Modelling ............................................................................................................................ 39
2.4) Information Paradigm ........................................................................................................................... 41
2.5) Enterprise Engineering Approaches - State-of-the-art ........................................................................ 42
2.5.1) Modelling Methods and Languages ............................................................................................... 46
2.5.2) Modelling Tools and Techniques .................................................................................................... 49
2.5.3) Standards ......................................................................................................................................... 50
2.6) Component-Based Approaches ............................................................................................................ 51
2.7) Review and Discussion .......................................................................................................................... 52
Summary ...................................................................................................................................................... 56

Chapter 3 : Research Approach and Objectives ............................................................................................... 57
3.0) Introduction .......................................................................................................................................... 57
3.1) Manufacturing Cell Design and Control Problems ............................................................................. 57
3.2) Research Approach - The Primary Vision ............................................................................................ 61
3.3) Research Objectives ............................................................................................................................. 63
3.4) Research Domains ................................................................................................................................ 64
3.5) Anticipated Benefits ............................................................................................................................... 65
Summary ...................................................................................................................................................... 65

Chapter 4 : Design and Reconfiguration of Manufacturing Cells - The Research Scope ................................... 66
4.0) Introduction .......................................................................................................................................... 66
4.0.1) Bottom-Up Approaches .................................................................................................................. 66
4.0.2) Top-Down Approaches ................................................................................................................... 67
4.1) The Research Concept ......................................................................................................................... 68
4.2) Adopted Methodologies ....................................................................................................................... 71
4.3) The Research Plan ................................................................................................................................ 73
4.5) Assumptions and Definitions ............................................................................................................... 75
Summary ...................................................................................................................................................... 78

Chapter 5 : Developing Semi-Generic Modelling Concept for Manufacturing Cells ............................................ 79
5.0) Introduction .......................................................................................................................................... 79
5.1) Semi-Generic Modelling Concepts ..................................................................................................... 79
5.2) Generic Task and Cell Groupings ....................................................................................................... 82
5.3) Conceptual Design of Semi-Generic Model ......................................................................................... 84
5.3.1) Modelling Functional Requirements ............................................................................................... 85
5.3.2) Modelling Information Aspects ....................................................................................................... 91
5.3.2.1) Information View of the Semi-Generic Model ............................................................................. 92
Chapter 6: A Component-Based Approach to the Modelling of Manufacturing Cell Control (MCC) Systems

6.0) Introduction

6.1) Detailed Design of the Semi-Generic Model

6.2) Designing Functional Aspects

6.3) Designing Information Aspects

6.3.1) CIMOSA-EXPRESS Mapping

6.3.2) Design of a Semi-Generic Information Model

6.4) Further Development of the Component-Based Approach

6.5) Resource Allocation and Configuration Method (RACM)

6.5.1) Review Resource Availability

6.5.2) Dividing Cell into Independent Groups

6.5.3) Meeting Functional Entities Requirements

6.5.4) Meeting Business Entity Requirements

6.5.5) Establishing Relationships between Cell Entities

6.5.6) Attribution of Procedural Rules

6.5.7) Services and Supportive Systems

6.6) Implementation Issues at the Semi-Generic Level

Chapter 7: Development and Use of Modelling Support for a Particular Cell

7.0) Introduction

7.1) Modelling Real Cases

7.2) Modelling Particular Cells - The Methodology

7.3) Generation of Particular Models

7.4) Development of Prototype Application Tools

7.4.1) Semi-Generic Modelling Tools

7.4.2) Cell Design Tool

7.4.3) Cell Application Tools

7.5) Infrastructure Services for the Prototype Tools

7.5.1) Supportive Integration System

7.5.2) Supportive Information System

7.6) Modelling Support for Operational Processes

Chapter 8: Case Study

8.0) Introduction

8.1) Description of the Case Study Cell

8.2) Use of the Modelling and Model Enactment Environment

8.3) Worked Example

8.4) System Reconfiguration and Extension - An Appraisal

Chapter 9: Analyses and Conclusions

9.0) Research Review

9.1) Discussion of Research Achievements

9.1.1) Analysing the Case Study Results

9.2) Analysis of the Change Capability

9.3) Final Discussion and Conclusions

9.4) Recommended Future Work

Appendix A: Library of Semi-Generic Building Blocks

A.1) Reusable Business Entities

A.2) Reusable Function Entities

A.3) Semi-Generic Information Entities

A.4) Reference Procedural Rules

A.5) Particular System Elements - The Default Configuration

A.6) Reusable Software Components

Appendix B: Sample Templates and Procedural Rules

Appendix C: EXPRESS Representation of the Information Model
List of Figures

1) Figure 1.1: Needs for migrating to the agile manufacturing systems ............................................ 20
2) Figure 1.2: Model of change in a manufacturing organisation and its impacts on the system design (Constructed using [1] and [18]) .................................................. 22
3) Figure 1.3: Supportive design approaches to map between conceptual level and operational level of manufacturing system design ........................................................................................................ 27
4) Figure 2.1: The CIMOSA modelling approach ............................................................................. 44
5) Figure 2.2: The CIMOSA functional modelling constructs ............................................................... 45
6) Figure 3.1: Part of the internal and external cell problems ................................................................. 57
7) Figure 3.2: The overall research approach .......................................................................................... 62
8) Figure 3.3: Potential applicability area of the research within the CIM domain (Y diagram adopted from [63]) ..................................................................................................................... 64
9) Figure 3.4: The areas of enterprise modelling covered by this work .................................................. 64
10) Figure 4.1: Overview of the concept developed in this research ...................................................... 70
11) Figure 4.2: Overview of the Research Plan ....................................................................................... 74
12) Figure 5.1: Conceptual Design part of the semi-generic manufacturing cell control model ............... 80
13) Figure 5.2: Task grouping concept in the manufacturing cells ........................................................... 83
14) Figure 5.3: Modelling stages at the CIMOSA requirement definition modelling level ................. 85
15) Figure 5.4: Semi-generic MCC model, 'Context Diagram' (See Appendixes E and F) ....................... 86
16) Figure 5.5: MCC model, 'Structure Diagram' for Task Management domain process ....................... 88
17) Figure 5.6: 'Behaviour Diagram' for the New Order business process ............................................. 90
18) Figure 5.7: 'Function Diagram' for the New Task Management domain process ............................... 91
19) Figure 5.8: The CIMOSA information model and semi-generic information model developed in this research (left part of the figure was adapted from [152]) ............................................................... 93
20) Figure 6.1: Developing the semi-generic model at the "detailed design" level ................................... 98
21) Figure 6.2: EXPRESS-G representation of the semi-generic information model at the schema level (for complete illustrations see Appendix C) .............................................. 105
22) Figure 6.3: Development and deployment of the Component-based approach within the CIMOSA modelling structure ................................................................................. 107
23) Figure 6.4: The concept of RACM approach to reusing activity components ............................... 109
24) Figure 6.5: Overview of the Resource Allocation and Configuration Method (RACM) ...................... 111
25) Figure 6.6: Resource availability and grouping the cells .................................................................. 112
26) Figure 6.7: Allocation of business and function entities to the cell groups ....................................... 113
27) Figure 6.8: Assigning resources to a group ...................................................................................... 114
28) Figure 6.9: Implementation issues at the semi-generic modelling level .......................................... 116
29) Figure 7.1: Model of manufacturing cells at the particular level ...................................................... 122
30) Figure 7.2: Developing a particular model for a sample PCB assembly cell at different level of modelling, based on the use of semi-generic model and prototype application tools .................................................. 125
31) Figure 7.3: Overview of the "Prototype Toolset" developed in this study ........................................ 128
32) Figure 7.4: The structure of Cell Design Tool .................................................................................. 130
33) Figure 7.5: Overview of information system developed for the particular model ............................. 135
34) Figure 7.6: An example for information transaction mechanism using the integration system .............. 136
35) Figure 7.7: Supporting operational changes by the use of particular models .................................... 137
36) Figure 8.1: Test case production line to produce high-pressure hydraulic cylinders ......................... 142
37) Figure 8.2: The requirement definition level of the particular model, designed for the test case ........ 143
38) Figure 8.3: Allocation of the modelling entities to the groups of the Machining Cell ........................................ 146
39) Figure 8.4: Part of the EXPRESS information model developed for the "Machining Cell" ................ 147
40) Figure 8.5: New layout of the "machining cell" with additional inspection units ................................ 150
41) Figure 9.1: Part of the common changes in manufacturing system studied in this research .................... 158
42) Figure A.1: The level of functionality for designing the software components ............................... 177
43) Figure B.1: Examples of SEWOSA Templates ............................................................................... 180
44) Figure B.2: Example of the Procedural Rules .................................................................................. 181
45) Figure C:1: Graphical illustrations of the EXPRESS model developed for the semi-generic information model (including 8 diagrams) .......................................................... 182
46) Figure D.1: Group application tool ................................................................................................. 192
47) Figure D.2: Supervisor application tool ............................................................................................ 193
48) Figure D.3: Initial interface of the Prototype CDTool developed in this research .............................. 194
49) Figure D.4: Resource Allocation Window of the CDTool to specify the entity relationships ............... 195
50) Figure D.5: Different examples of specifying the entity relationships ............................................. 196
51) Figure D.6: The final system entity configuration ........................................................................... 197
52) Figure D.7: The CATool for group Supervisor ................................................................. 198
53) Figure D.8: Sample CATool for groups ............................................................................ 200
54) Figure E.1: Part of the 'Object Diagram' developed for the semi-generic model using the SEWOSA CASE Tool (for complete illustration see appendix F) .................................................................... 208
55) Figure E.2: Overview of the graphical representation of the MCC model at the detailed design level, developed using the SEWOSA CASE tool ................................................................. 208
56) Figure E.3: The execution level of the MCC model developed using the SEWOSA CASE tool ... 210
57) Figure F.1: Graphical representation of the semi-generic model (including 34 diagrams) ........ 212
58) Figure G.1: Modelling design considerations .................................................................... 238
59) Figure G.2: The CIMOSA modelling framework (adopted from [184]) ............................... 244
60) Figure G.3: CIMOSA business modelling (adopted from [184]) ........................................ 244
61) Figure G.4: Domains with interacting Domain Processes .................................................. 246
62) Figure G.5: Decomposition of a Domain Process into Business Processes and Enterprise Activities . 246
63) Figure G.6: Domain Process Representation as a network of Enterprise Activities ............... 246
64) Figure G.7: Enterprise Activity with Inputs and Outputs ..................................................... 246
65) Figure G.8: Functional Operations executed by Functional Entities (Resources) ................... 246
66) Figure G.9: GERAM Framework ....................................................................................... 249
67) Figure G.10: Functional blocks of CIMBIOSYS system .................................................... 250
### List of Tables

1) Table 5.1: Methods and tools used to develop the semi-generic model of MCC systems .................... 85
2) Table 5.2: Some of the semi-generic information entities defined at the requirement definition modelling level .................................................................................................................................. 94
3) Table 6.1: Mapping from CIMOSA information view to the EXPRESS language structure .......... 104
4) Table 6.2: Information requirements in EXPRESS language format (see also table 5.2) .............. 104
5) Table 8.1: Resource classification for the “hydraulic cylinder machining” cell ........................ 145
6) Table 8.2: Populating the generic function entities with the test case data .............................................. 145
7) Table 8.3: Allocation of the business entities to the Machining Cell Groups ........................................ 145
8) Table 8.4: Modelling configurations required for accommodating the “machining cell” changes .. 150
9) Table 9.1: Summary of research achievements aimed at improving “change capability” .......... 163
10) Table 9.2: Summary of the research result ...................................................................................... 171
11) Table A.1: Library of the Building Blocks - Reusable Business Entities ........................................ 172
12) Table A.2: Library of the Building Blocks - Reusable Function Entities ........................................ 173
13) Table A.3: Library of the Building Blocks - Reusable Information Entities (including 5 tables) ... 173
14) Table A.4: Library of the Building Blocks - Reference Procedural Rules ...................................... 176
15) Table A.5: Library of the Building Blocks – Software Components Descriptions ........................ 179
16) Table D.1: Computer tools implemented in this research .............................................................. 190
17) Table G.1: Modelling Framework Comparison - Life cycle (Modelling Levels) – adopted form [72] ......................................................................................................................................................... 252
18) Table G.2: Modelling Framework Comparison - Model Views– adopted form [72] ...................... 252
19) Table G.3: Modelling Framework Comparison - Generality ............................................................. 252
## Glossary of Terms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>Business Entities</td>
</tr>
<tr>
<td>BP</td>
<td>Business Process</td>
</tr>
<tr>
<td>CATools</td>
<td>Cell Application Tools</td>
</tr>
<tr>
<td>CDTool</td>
<td>Cell Design Tool</td>
</tr>
<tr>
<td>CIMOSA</td>
<td>CIM Open Systems Architecture</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>DP</td>
<td>Domain Process</td>
</tr>
<tr>
<td>EA</td>
<td>Enterprise Activity</td>
</tr>
<tr>
<td>EO</td>
<td>Enterprise Object</td>
</tr>
<tr>
<td>FE</td>
<td>Function Entity</td>
</tr>
<tr>
<td>FO</td>
<td>Function Operation</td>
</tr>
<tr>
<td>IE</td>
<td>Information Elements</td>
</tr>
<tr>
<td>IO</td>
<td>Information Object</td>
</tr>
<tr>
<td>MCC</td>
<td>Manufacturing Cell Control</td>
</tr>
<tr>
<td>OV</td>
<td>Object View</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>PR</td>
<td>Procedural Rules</td>
</tr>
<tr>
<td>RACM</td>
<td>Resource Allocation and Configuration Method</td>
</tr>
<tr>
<td>SC</td>
<td>Software Component</td>
</tr>
</tbody>
</table>

1 The above terms have been repeatedly used throughout this document. However, the reader can refer to the “Index” section at the end of this thesis for a complete definition of terms.
Style of Presentation

For the sake of clarification the following conventions have been used throughout this document:

1. All terms introduced and used by the CIMOSA modelling architecture and adopted in this document are formatted in *Italic* style.
   
e.g. *requirement definition* modelling level

2. All terms introduced and used by other architectures, standards, methods, techniques and research projects (e.g. SEWOSA, EXPRESS, and STEP) and adopted in this document are written within single quotation marks.
   
e.g. SEWOSA 'context diagram', EXPRESS 'attribute'

3. All terms defined in this study are specified in quotation marks.
   
e.g. "allocated procedural rules", "software component template"

4. Sections, Figures and Tables numbering styles start with chapter or appendix numbers.
   
e.g. §6.5.6 in chapter 6, §G.2 in appendix G

5. In this thesis the acronyms are used in form of plural by adding suffix of 's and should not be confused by possessive apostrophe.
   
e.g. EA's and PCB's
Chapter 1  

Business Trends and their Impact on the Requirements of Manufacturing Systems

1.0) Introduction

Since 1970 the business world has dramatically changed. The magnitude of this change may be compared with the appearance of computers in industry during the 40s and 50s or even the industrial revolution at the end of 17th century [1]. The business world needs to develop methods and techniques suitable for an environment of market globalisation, customer orientation, and knowledge capitalisation [2]. Today businesses are operating under uncertainty arising from: governments, which introduce new environmental, social, and industrial regulations [3]; marketplace which is driven permanently by change in customer requirements [4]; uncertainty in capital markets which orient the overall market direction; an advancing base of technology [5]; and competition which is tougher than ever before [6]. These pressures continue to require businesses to change, where commutative pressures of different types result a very significant effect on the businesses.

The history of commerce shows that when disparate pressures align, they can cause a permanent step change in businesses conditions [1]. For instance, international trade was once based in Venice. The British Empire took over for a long period of time and then the benchmark became major international companies, mostly American and Japanese and more recently Asia pacific in origin [6].

Monopolies still occur in some business sectors such as on a regional basis in regard to oil trusts and communication corporations. Following governmental support for free markets and foreign investment, many regional markets have evolved into global ones, with new competitors threatening long lasting monopolies [6]. Currently successful business is often a matter of price, quality, reliability, and customer acceptance. Issues connected with investors or market locations generally have a reduced influence.

Today trade experts can better predict and analyse the impact of business changes than before. However, often the exact nature of changes is not clear to anyone, and even less obvious may be the probable effect of such changes. It is said that the world markets are currently undergoing a process of great change [1]. In the future therefore it may become less likely that a single company will dominate a market sector. Only companies with novel products or services can expect to survive in a climate of strong international competition [7]. In many cases companies will find it challenging just to remain in business, let alone sustain growth. Market information will be a major, possibly prime resource [1]. Attracting and realising customer requirements will be the major concern by any possible means such as social, industrial or even political and military.
pressures.

These factors reveal the crucial needs to remain aware of international market trends and a necessity for businesses to be flexible, ready-to-change and have a capability to cope with unanticipated changes as they occur.

1.1) Needs for Change Capability

Thus as we approach the twenty-first century, radical changes are taking place that are transforming industry perspectives. In many industrial sectors (if not most) the marketplace has become global [2]. Yet customers expect to be treated individually and consequently may require small quantities of customised products. For a manufacturer this implies lower volume production, and more varied products, often with complex and costly quality and service requirements. Many companies have developed a capacity to deal with a wider range of products with shorter product development and production lead times [8]. Realising such a capability has caused new and difficult challenges for companies [9].

In the 1970s and 80s many companies tried to improve their control of manufacturing operations through the use of concepts such as: CIM (Computer Integrated Manufacturing); computer assisted production and materials planning (based on systems like MRPII - Manufacturing Resource Planning). This was achieved with varying degrees of success [8].

In the 1980s and 90s world class manufacturing and marketing concepts have been adopted including: TQM (total quality management) methods for controlling processes and continuous improvement [10]; just-in-time manufacturing with cellular manufacturing [11], quick change-over, Kanban [12], and zero inventories [13]; team-based continuous improvement with self-directed work teams [8, 14]; and agile manufacturing [15].

Today, in most market sectors, mass production methods have become obsolete. Seldom are they profitable due to the problems with large inventories and excessive overheads [8, 16]. In a similar situation a small or medium sized enterprise may prevail over large companies as a result of its ability to respond rapidly to changing requirements and operating conditions. In such cases, Make-to-Order capabilities may be essential to maintain competitiveness [2].

To be competitive companies will often need to respond rapidly to customer demand, possibly by bringing products to market more quickly than ever before. Increasingly therefore the key business processes of an enterprise need to be implemented in a way that promotes competitive behaviour whilst positively accommodating change in the business environment [17].

Therefore as business conditions change, the business strategies and goals of an enterprise need to be developed and refined. Changes in these conditions need to be reflected in changes to the
configuration of an enterprise so that it remains competitively aligned to external requirements, conditions and constraints. To realise better alignment, as appropriate various parts of an enterprise will need to expand, consolidate, form partnerships, diversify, or geographically relocate. This may require significant organisational and structural change, such as the centralisation or decentralisation of functions and growth, or shrinking or outsourcing of business units [18].

However, in general, contemporary enterprise systems will have not been designed in such a way that they positively enable and support change. On the contrary, it is reported that the lead-times associated with significant system change are commonly of the order of nine months to five years [18]. Such delays are unacceptably long, as over that time frame, business and environmental conditions are likely to have changed appreciably so that new business and associated production requirements exist. Inevitably this will result in inefficiencies and potentially lost market share (or worse) if competitors can respond more rapidly.

It follows that in a climate of persistent market change many businesses will need to be capable of "responding to" and/or "accommodating" change, so that they are capable of behaving competitive under product market, social, political, economical and their environmental conditions that are increasingly globalised, volatile and complex in nature.

Arguably "change capable" manufacturing brings nothing fundamentally novel to the ‘table’, as various initiatives world-wide have advanced flexible manufacturing concepts [17]. However a focus on "change capability" emphasises the fact that increasingly contemporary manufacturing enterprises require ‘the ability to thrive in an environment of rapid and unpredictable change [17] and therefore either they must be able to rapidly respond to change in requirements, or be capable of accommodating changes in operating conditions.

Based on this line of reasoning the research reported in this thesis is founded on the assumption that a “responsive” or “change capable” enterprise will in general require manufacturing systems that are “responsive” and/or “change capable”. It develops this line of reasoning by researching, designing, implementing and testing a new (so called “component-based”) approach to producing “responsive” and “change capable” manufacturing systems from flexibly organised sets of interoperating human and technical resource elements that collectively have a capability to realise the business processes of an enterprise. It follows that the overall aim of this “component-based” approach to the life-cycle engineering of manufacturing system is to enable manufacturing

1 The term 'enterprise' is used here in a generic sense, encompassing companies (and groups of companies) that realise products and provide services, be that in industrial, commercial, financial, educational or government sectors.

2 The term ‘manufacturing’ is used in a generic sense to encompass the producing of products (be they aeroplanes, ships, computer games or mass produced consumer items) and supplying of services (be they financial, educational, consultancy, etc).

3 Here the term resource is used to denote an entity (human or technical) which can play a role in the realisation of certain activities and tasks, when it is available [19].

4 Processes constitute "a sequence (or partially ordered set) of enterprise activities (linked by precedence relationships), execution of which is triggered by some event, and will result in some observable or quantifiable result" [19].
businesses to cope with complex and uncertain conditions by aligning the operation of their manufacturing system to changing business goals and requirements on an ongoing basis.

By so doing, a related aim of this new approach is to stabilise the operation of an enterprise despite disturbances arising from a turbulent environment.

Recent published literature on "agile" systems has emphasised that agile or change capable organisations are unlike lean or world class manufacturing organisations which are good at doing the things, which can be predicted and controlled. Agility has been defined as:

"Ability to thrive and prosper in an environment of constant and unpredictable changes [8]."

"An ability to constantly reconfigure strategies and processes and to examine market positioning; as external conditions will change [5]."

It follows that the property of "agility" can be of strategic value to an organisation, not only to accommodate change but also to respond to new opportunities within an unstable market environment. However, this property may be of greater value in some industrial sectors than in others. In the food industry, consumer electronics and automotive industries the need for agility is readily apparent [20]. New products should be produced rapidly to maintain the competitive position of a product range. It may be that a continuation of such trends will result in further fragmentation and specialisation of markets [15]. Under such conditions, we may expect that innovative and agile companies will be best able to survive.

Apparently therefore, current industrial trends mitigate against old ideas of huge factories making massive quantities of a relatively small number of standard products. Furthermore such pressures signal a need for new forms of manufacturing system that are capable of rapid and effective configuration to provide operational behaviour that can be modified frequently in a yet to be specified manner. Figure 1.1 depicts in concept a growing need to migrate toward change capable manufacturing.

In summary, there is a growing requirement for dynamic and re-configurable manufacturing systems that organise and operationalize human and technical resources so that they realise core business processes in a timely, effective and efficient manner [18].

1.2) Need to Deploy a New Approach to System Design and Reconfiguration

Present day manufacturing systems do not meet sufficiently well the business needs of many manufacturing companies [21]. Direct evidence for this assertion has been provided at various industrially led forums and by studies of industrial practice funded by governments [22]. In many industrial sectors and companies it is claimed that a significant gap exists, between the operation of systems and the business process requirements they support. Although it is seldom practical to quantify the effect of such a misalignment in business terms, it is known that this has led to unprofitable, even uncompetitive behaviour [21]. It is evident that the magnitude of this
misalignment may vary over time in a complex and unknown manner. That will depend upon the nature of the various specific manufacturing operations that need to be performed.

Thus, many senior managers in industry and commerce believe that, as global competition grows, increasingly it will be necessary to more "sharply tune" the configuration of any highly successful enterprise [23, 24]. However, any realignment of system operations should reflect strategic goals and activities. This may include: innovative response to market research on customer needs; maintaining competitive behaviour while manufacturing according to specific needs; realising unforeseen customer needs; co-operation with other firms; achieving the rapid distribution of products and services; and achieving re-configurability (i.e. fitness for system change) of products, resources and organisation structures [14, 25].

Market requirements for configurability and flexibility may best be met by establishing short and medium term business partnerships [17]. This can be achieved by utilising electronic trading methods [26, 27]. Implicitly however business partnerships co-ordination and co-operative operation will cross boundaries of companies' responsibilities and systems. This in turn implies a need to deploy integration methods, standards, and techniques, which can be used across those boundaries [28]. To achieve the configurability and flexibility, next generation enterprise systems need an inherent capability to be 'retuned' (by whatever process and means) so that the overall organisation can remain stable whilst adjusting to change [18]. In many cases, product design
activities will need to be closely integrated with process activities and the resources available to an enterprise. The need for fast and effective design means that the traditional approach to having all new products routed through a design area may prove ineffective. Such practices may cause delay, misunderstanding, and a lack of co-operation between the design area and the production floor [8]. A natural conclusion is that the design process must be integrated with the manufacturing process [2, 29].

Products may be modularised to facilitate their own configuration [30]. This can simplify and even negate the need for separate design processes for each product. Likewise, operational processes and the systems and resources, which realise them, can be modularised. This can facilitate the process of enterprise design and reconfiguration by providing means of realising improved alignment between goals and enterprise operations more quickly and with reduced engineering effort [29]. Hence competitive behaviour of systems can be realised more effectively by utilising well-defined and reusable modularised system resources, and by deploying suitable methods, standards and techniques to integrate these modules in a way that improves the co-ordination and control of enterprise processes (and thereby their underlying operations, tasks and activities).

Therefore as a result of using

1) a well-defined and suitable set of enterprise system modules (that will also be referred to in this thesis as enterprise resources or system components), and

2) a methodological and computer assisted approach to rapidly and effectively configuring (and reconfiguring) enterprises (and their constituent manufacturing systems) from organised and interoperating groups of system modules (i.e. resources and components) in alignment with business requirements

it should prove possible to improve the responsiveness of an enterprise and accommodate various kinds of unforeseen change [18].

1.2.1) Model of Change

It follows that in many (if not most) cases, modern manufacturing organisations need to constantly improve their business plans to maintain or enhance their competitive position [9, 25]. They need to be able to accommodate market and other environmental changes, which arise from factors such as: new customer demands; new technology; new environmental regulations; new challenges; etc.

To develop improved business plans in alignment with market demands, a semi-formal method is required to structure innovative activities carried out by team of people responsible for recognising and characterising those demands and to ensure that supportive reactions will occur in different parts of the organisation [4]. Morris [1] has defined four different stages involved in formalising
change in organisations. Figure 1.2 depicts these stages. They include:

- Market assessment
- Reengineering business
- Impact of change
- Implementing change

**Market assessment:** An organisation should continually assess the condition of the company in comparison with the market demands and the performance of other competitors. A correct prediction of market demand and appropriate adjustments to business plans are the key to success. Furthermore, a market encouragement policy may persuade customers to demand what the organisation can provide.

**Reengineering business:** When market demands change, the organisation’s plans and objectives may be compromised and the organisation’s capacities need to be reconfigured to realise an improve fit, to the new market situation. At this stage, it is essential to realise what exactly should be changed or adjusted and how such change can affect different parts of the organisation. In this case, new business process model(s) may be required.

**Impact of change:** When the required business changes have been understood, the impact of change needs to be managed and co-ordinated to reduce negative effects on the different

---

5 In this context the term “semi-formal” is used to imply the use of formal modelling approaches to provide structure and analytic support for human decision and action making.
departments of the organisation and on their staff. Thus work flow, marketing strategy, financial and business rules and personnel regulations need to be revised as well making changes to production technology, which should be changed to cover new operational requirements. The technological impact may result in changes to: resource configuration; operational processes; use and flow of information; and instructions defining detailed technical operation and integration of systems.

**Implementing changes:** At this stage, new plans need to be realised by implementing the new conditions and designs. The new operational processes and information systems should be used in accordance with new business rules and new technical instructions and methods. A fundamental issue during the realisation of new processes and systems by the organisation is the ability to be able to flexibly match new designs to new situations [15]. Furthermore, the results of an implementation process associated with any change needs to be closely evaluated and fed into higher levels of decision making to revised the changes and their impact as required.

The model depicted by figure 1.2 represents the change process associated with manufacturing systems from an organisation point of view. Whereas, from a technical prospective the effect of implementing changes (especially technological ones) should be considered when designing responsive and change capable systems. It should be possible to determine the capabilities required from enterprise resources and to match these capabilities to the new production and services needs. Hence implicit in the ability to reconfigure systems and reassign resources is that the systems used should have an inherent capability to realise the new requirements associated with any given change [17, 31].

It follows that benefit should be gained by classifying different types of change with respect to their business and technical implications. This should allow the designer and implementers of enterprise processes, systems, and resources to develop more agile solutions. However it is by no means a trivial exercise to develop such a classification in a generally useful form. Clearly for an entity as complex as a manufacturing organisation competing in world markets, change can arise from various sources and can influence the design of an enterprise and its manufacturing systems in numerous ways, e.g. in terms of the nature, scope, magnitude and time-scale of its impact. However to place some bounds on the likely impact of change on manufacturing systems in this research the initial focus of study was on the impact of:

1) Change in Business Requirements

2) Change in Production (or Service) Requirements (and associated detail operational changes)
In the context of its general impact on manufacturing systems, change in business requirements has been considered and formally linked to requirements for system(s) reengineering, where that change will be expressed in terms of some business process change. It has been assumed that change of type (1) may involve a conceptual redesign of systems (in terms of their business goals, functions they must perform, the nature and number of resources they deploy and the way in which they should be organised) and may lead on to major system(s) reengineering in the event of a given change or accumulation of changes. Whereas it was assumed that change in production (or service) requirements will typically require system(s) reconfiguration [28, 29]. Generally speaking, it has been assumed that the impact of type (2) change will be more constrained in its scope and impact, particularly if a specific change of that type had already been anticipated in the original system design. In certain respects, type (2) change is akin to notions of flexibility, as defined by Dove [32]. Hence a requirement for technical flexibility will typically imply a capability to facilitate change of type (2), i.e. inherent ability to be able to readily change the ways that system resources interoperate to achieve a new set of operational requirements, where these new operational requirements may well have been established as part of a process of reengineering.

It was also assumed that change of type (2) can be expected to require sufficient changes to the use of resources (e.g. people, machines, and information) to be carried out before operating associated systems. In a traditional bottom-up approach to system design, it will be necessary to determine the physical circumstances under which an operational section of the organisation (e.g. a manufacturing cell, or shop floor) will operate, and to specify the operational design (e.g. process plan and factory layout). Here, the definition of process might determine what physical resources and production methods should be used. Such a bottom-up approach, although apparently sensible, can cause problems in terms of linking it to business and conceptual design requirements, especially in climate of ongoing change. For example, consider the case of designing a process, which is analysed from the bottom-up, starting with the definition of physical operations up to determining what resource, service, and business requirements must be met. When product and service requirements change occurs, possibly initiated by external factors (see impact of change in figure 1.2), a system designer can only compare the new conditions with needs of similar operational processes, modify the process accordingly, and compare the result again. This repetitive trial design method will introduce delay before the design and operation of the system become effective. Even then there is no ways of knowing if by achieving a local optimisation, the result will be some corresponding impediment on the overall business performance of an

---

6 A defined focus on capturing requirements change in the form of a change to a business process model is consistent with recent literature (that recommends a focus on value-added processes that cross organisation boundaries), the availability of commercial process modelling tools (that can formally describe changes in behaviour and activities) and a centre of current research interest in the MSI Research Institute at Loughborough University.

7 It should be noted that configurability might differ from flexibility in different classes of manufacturing system. Here the former implies the need to cope with greater scale of business change and that the nature of that change may be less predictable. Whereas the later property may provide a capability to improve the technical adaptability of the operational system.
enterprise.

Whereas if a top-down approach to enterprise design and implementation is taken, the designer should start by considering objectives of the enterprise as a whole and thereby define the system requirements conceptually, in terms of possible production methods and organisational arrangement of resources which could be compared and a chosen solution developed to realise the objectives. As a consequence of using top-down approach, the behaviour of the system, can be described in the form of a conceptual model, or so called business model [19], which can be used to determine functionality and behaviour of system components [33]. An inherent property of using a top-down approach to enterprise design and construction is that changes in business requirement can be reflected in well-defined changes to system behaviour and resource requirements [34, 35]. However, problems arise as the development of conceptual models implicitly require processes of abstraction to aid the development of conceptual understanding by system designers. This may well lose detailed information required by system implementers and those entities responsible for the physical operation of systems.

Hence, some means of maintaining consistency between conceptual designs and real systems need to be achieved, so that conceptual enterprise models can be detailed to produce more exact models of enterprise resources and the way in which their operations need to be integrated. However, it is generally understood by the enterprise modelling and integration research community that there remain practical difficulties of maintaining consistency (and thereby alignment between conceptual design and alternative physical solutions) in a way that can be generally applied [36]. Until solutions can be found to such problems the use of design models to structure and control the implementation and operation of enterprise system will be limited.

It follows that there may well be a good case for representing (or modelling) and maintaining an intermediate level of design detail to interpret and update models generated during conceptual design so that they remain in alignment with the operational system. In addition, this could allow the results of operational processes to be reflected in better (more accurate and complete) models that can be used at the conceptual level. Based on this understanding, this research will investigate ways of providing such as intermediate level of modelling and of making these models operational, so as to bridge the gap between current top-down and bottom-up system engineering and system configuration approaches, and therefore between system design and system realisation (see figure 1.2). The intermediate level of modelling should provide a flexible and widely applicable way of mapping changes in business requirements onto changes in system interoperation.

1.3) Common Problems Associated with Current Manufacturing Systems

To date, generally contemporary manufacturing systems are more flexible in respect to production requirements change (change type 2) than business requirements change (change type 1) (see figure 1-2) [17, 23]. This is most probably because batch manufacturing techniques have been advanced
over a number of decades and changes of type (2) tend to be more predictable and less extensive in nature. Whereas current generation manufacturing systems do not readily support change of type (1). This inflexibility results as a direct consequence of the concepts, methods and techniques currently deployed during system design and construction which result in an "inflexible mapping" between conceptual requirements (of what a system should be doing, in terms of tasks, operations and activities) and the way in which a system operates (and hence the way in which it organises and controls the collective behaviour of enterprise resources) [17].

As a consequence new approaches to system design and construction are required that flexibly map between conceptual requirements and system operation and thereby can more readily support change of types (1) and (2). Such a capability is in the long term interests of both ‘manufacturers’ (end user of systems) and vendors (of systems) [18]. Manufacturers will be able to realise better alignment of systems with business and production goals, whereas vendors will be able to more readily reengineer and reconfigure their systems for use at different customer sites. This would lead to better "tuned" enterprise systems and more effective reuse of resources (and hence system components). Naturally the degree of specialism of systems and resources will tend to increase as their level of inherent complexity grows, and indeed as their scope (and granularity) increases there may be less distinction between the two (i.e. between resource and system). Some product design systems, production planning and control systems, product introduction systems and materials handling systems comprise software packages that have been conceived and constructed with a view to their reuse in different host enterprises. However invariably they will have been engineered in conformance with a particular specification that explicitly details how a particular set of operations, tasks and activities should be semi-automated. Some flexibility may be crafted into the system but normally this will be done in a way that allows only limited support of certain types of change. Indeed it is likely that only those types of change that can be predicted in advance will be supported adequately. To further complicate matters it is also important to bear in mind that the cost of unnecessary flexibility (and associated capability) can be high, both in terms of increasing life-cycle cost and maintaining the runtime performance of a system [18].

1.4) Possible Solution
The means available to design and construct manufacturing systems have advanced significantly over the past decades. New design methods and technologies are facilitating the development of more flexible manufacturing systems. Technologies which promise new approaches to the life cycle engineering of flexible manufacturing systems include: Enterprise Modelling Concepts; Systems Integration, concepts and technologies; Component Technology; etc. (see figure 1.1). However outstanding and complex problems, such as the gap between high level conceptual design and operational design discussed earlier, remain to be solved before these enabling technologies can be unified to provide more generic manufacturing system that can respond to and accommodate the kinds of change that commonly occur in modern manufacturing enterprises.
As illustrated by figure 1.3, there are at least three groups of design issues that need to be addressed to bridge the gap between the conceptual design level and the operational level of manufacturing systems.

These groups centre on the:

- Resources provided by manufacturers
- Software components supplied by vendors
- Models, methods, concepts used by designers

The physical resources (e.g. operator, machine, and know-how) utilised by manufacturers, collectively constitute the capabilities of the system. The inherent flexibility, capabilities and capacity of these resources can directly improve the responsiveness and ability to accommodate system change under specific operating condition.

Computer application tools (that will also be referred to in this thesis as components) provided by various vendors, contribute significantly to the control and co-ordination of system operation and resources and can be deployed in semi or fully automated scenarios.

Generally, these tools are built to support a class of activities commonly required to be organised and constructed by enterprise systems (e.g. for scheduling activities). Whereas the concept embodied in the development of component technologies is that it is possible to provide executable building blocks of software tools that can more closely be matched to the required functional building blocks of systems, and hence to specific competitive behaviour requirements of a particular enterprise. By having a suitable set of components from which to choose it may be possible to configure them in different ways to meet a wide range business and operational requirements of manufacturing systems, possibly even under conditions where unpredictable changes occur.

To investigate this possibility further, it is necessary to choose (or design) suitable concepts, methods, and techniques to realise and formalise business requirements, alternative system configurations and representation of operational systems in a meaningful and effective way. There is also a need to be sufficiently generic to adapt a wide range of business requirements, and be sufficiently flexible to support the detailed description of alternative set of real system components.
and resources.

Hence this thesis seeks to develop the premise that an improved approach to design of manufacturing system can be achieved by realising and developing an effective unification of the three set of issues as indicated in area (1) of Figure 1.3. It is assumed that such an approach can help to unify future design and construction efforts of manufacturers, vendors, and designers. Currently, vendors have difficulty when building application tools that conform closely to the original concepts and methods deemed to be necessary by system designers (i.e. area (2)). Also designers do not have detailed knowledge about requirements and constraints of vendors (even when they use standard methods). Furthermore, common communication and integration concepts and mechanism should be shared by software tools and physical resources, needed to build interoperating systems in a rapid and reflective way (i.e. area (4)). It is necessary to deploy consistent interaction techniques to achieve communication amongst system resources. Although normally physical resources and software applications will constitute functional building blocks of the systems, they may also provide a service (that is shared by system components and resources). There is a need for models, methods, and concepts to facilitate the use of these function blocks and services so that they facilitate the realisation of business requirements in an effective and flexible way. Area (3) delineates the methods used to formally describe resources and their configuration. These methods should ensure that responsiveness and ability to accommodate change are inherent properties of resultant systems.

It follows that the focus of this research is on realising an effective unification of methods and tools to cater for the three groups of design issue identified above. The main emphasise is on mapping between requirement definitions and organised group of resources (i.e. area (3)) and the software components that have a capability to operationalize resource sets in alignment with defined requirements (i.e. area (2)).

In this research area, to date a variety of formal methods and techniques have been developed, mainly to separately address one of the three groups of design issues depicted by Figure 1.3. Also these methods in as much that methods and techniques have been developed in a largely independent way where the underlying concepts that span different disciplines and may involve various vested interests (i.e. of businessmen, managers, sociologists, technologists, shop-floor personnel and their system and component suppliers). To date, the largely separated and fragmented use of such methods and techniques has not induced a significant step change in enterprise engineering practice [18]. Hence with the aim of contributing to the development of “change-capable” manufacturing systems, this study will investigate methods and techniques that support enterprise engineering and enterprise operation in a constant way, thereby facilitate (re)engineering and (re)configuration of manufacturing systems.
1.4.1) Modelling of Manufacturing Systems

Modelling can be viewed as representing the state of a real world event (or system) and so to produce formal description that can be reused in various ways [37]. For example use of a model should help in the prediction of system behaviour during its life cycle. Models of manufacturing systems, can be used to underpin methods [38-40] that:

- simulate conditions of a system, in support of system design,
- instruct and control the implementation of real world systems,
- monitor the status of system entities during system operation.

A valid and complete model can contain information that can be utilised to determine how a system can be used more efficiently. Thus, a modelling tool can prove extremely helpful, particularly if it is consistent with design concepts, tools, and system resources to support the life cycle of engineering.

Hence, bearing in mind the stated objectives of this research it was decided that a modelling approach is required to formalise and support the life cycle of manufacturing engineering systems. It was understood that the modelling approach chosen and developed should encompass issues related to:

- enterprise integration,
- supportive information technology;
- physical resources.

Therefore it was envisaged that modelling concepts and modelling tools would underpin a component-based approach designed to engineer and configure “change capable” manufacturing systems.

1.5) Research Boundaries and Requirements

Therefore this research study seeks to contribute to the development of an approach to the design and construction of “change capable” manufacturing systems. Resultant target systems should be able to respond to complex and uncertain changes in business and market conditions. Furthermore the research will aim to unify and extend the use of available systems modelling, integration and component methods and concepts to embed three classes of properties into change capable manufacturing systems, that are:

- re-configurable (§1.1) – to enable adoption and evaluation towards new conditions of the system as required,
- modular (§1.2) – to enable the realisation of reconfigurable systems from well proven components and resources,
- generic (§1.4) – to facilitate their effective application in a defined application domain that is broader than domains typically serviced by individual contemporary manufacturing systems.
A particular research focus is on the use of modelling methods to bridge the current gap between; conceptual design; and system operation (see Figures 1.2 and 1.3).

Hence it is necessary to review and analyse the literature on available modelling tools and to determine the nature of outstanding problems posed in a manner that informs the development of a new approach.

The application and the proof-of-concept testing of this research would need to be limited for reasons of resource constraint. In this case the domain of manufacturing cell systems was chosen to match the manufacturing needs of an industrial collaborator.

The primary outcome of this study was expected to be a definition (i.e. in terms of level, type, domain, etc.) of modelling methods that are appropriate in support of the life cycle engineering of manufacturing cells to improve the change capability embedded into their control system. Thus, a methodology for organising, capturing, and operationalizing manufacturing cells has been developed along with proof of concept models and tools that help to examine the validity of the approach and to prove the concepts.

As illustrated by figure 1.3, the research is limited primarily to a determination of relationships between the production methods and resources needed to improve the integrity of: 1) the conceptual design of manufacturing cells and 2) models of runtime control systems used within the cell designed.

Modelling methods, tools, and techniques will be studied and an appropriate set of these approaches selected, developed, and unified. These approaches need to be consistent with contemporarily thinking on software engineering tools and concepts. A number of candidate concepts, models, and software engineering tools that were considered will be reviewed in the next chapter. In addition, it was found to be necessary to develop new approaches to incorporate their use into subsequent phases of the research.
Summary

Business conditions are typically in a state of constant change and greater rates of change are predicted in the future. Therefore, a new generation of manufacturing systems is required to cope with change on an enterprise-wide state. Such systems need to be more responsive and able to accommodate change than ever before. Accordingly, the design and construction of manufacturing systems need to be revised and new support methods and tools need to be developed.

Next generation manufacturing systems should be change capable so that they can adapt and evolve to satisfy new business and operational requirements. However, they should not incorporate unacceptable redundant capabilities that may result in the high cost of system development and maintenance.

The research concept was defined as development of a new approach to the design of manufacturing systems at an intermediate level of abstraction to improve consistency between models used in high level conceptual design, and to describe the operation of system need to be overcome.

To develop such an approach, the actions and knowledge of system designers, vendors, and manufacturers should be harmonised to provide a flexible system design and reconfiguration methodology, capable of meeting changes in both business and operational requirements.

To restrict the research boundaries and make the project a manageable subject of PhD research, all experiments are carried out in the domain of manufacturing cell systems.
Chapter

2

Literature Review

2.0) Introduction

Market and environmental requirements are changing rapidly [2]. Therefore agile manufacturing enterprises are required, with a capability to respond rapidly to such changes [31]. A major difficulty experienced in the design of change capable manufacturing systems is one of achieving an integration between conceptual design activities (which may reflect customer needs) and operational system design activities (based on establishing suitable groupings of physical resource capabilities).

Therefore, methods are required to formalise definitions of business requirements as logical definition sets of activities that define ways of meeting these requirements via operational specifications that describe operational groupings of manufacturing resources and their capabilities. Ideally the approaches (and the methods and tools) used to design and configure change capable manufacturing systems should be generic, modular, and reusable (§ 1.5) and their use should be set within a wider context of meeting enterprise (i.e. global) requirements. In this regard, a need to deploy enterprise modelling approaches was identified (§ 1.4) to provide a framework within which manufacturing system design and reconfiguration can be achieved.

Therefore this chapter reviews:

- basic definitions related to manufacturing cells and common research concepts and assumptions made in respect of their development,
- modelling approaches and means of using models to engineer primarily large scale systems,
- the current state-of-the-art in relation to the system engineering concepts, methods, and tools available in this research area.

2.1) Manufacturing Systems – Definition of Terms

Manufacturing systems have been categorised in various ways. Major research in this area has been carried out by teams of researchers at NBS/NIST and their findings have been developed and exploited within CAM-I research projects [41]. As part of NBS/NIST studies a hierarchical decomposition of organisational levels in typical companies included: factory, shop floor, workcentre, cell, workstation, resource unit, and equipment levels. Descriptions of these classes have been developed with respect to (1) common activities that manufacturing systems carry out and (2) the facilities and devices commonly deployed at each level [28, 38]. By combining the use of these two types of classification, different physical layouts of factory systems can be categorised. With reference to these classifications the focus of this research is on the configuration and control of groups of devices operating at a specified physical location in a company, that are typically
managed by people according to some pre-defined work plan. Generally computerised equipment will be used to support humans, thereby to facilitate device control and to organise and control interactions within a system or in broader terms in conjunction with the other groups of devices operating with a host enterprise.

This research will consider groups of these devices and their control system elements to comprise "manufacturing cells".

2.1.1) Manufacturing Cells

According to the survey carried out by Wemmerlov [42], the majority of manufacturing cells have been developed within the metal-working industry such as: machinery and machine tools; agriculture and construction equipment; defence products; hospital and medical equipment; engine production. This mainly includes production, assembly and product inspections activities [42].

Luggen categorised manufacturing systems within four groups [40], namely:

- Traditional stand-alone NC machines
- Single NC machine cells (also called minicell)
- Integrated multi-machine cells
- Flexible manufacturing systems (FMS)

Common tasks carried out within a manufacturing cell have also been discussed by many researchers including Bauer [43], Camarinha-Matos [44], OMG [45], Creasy [46], Jones [47], VanBrussel [48], Barkmeyer [49]. These tasks have been categorised into 3 classes, namely: a) tasks involved in receiving and scheduling production within a cell, b) co-ordinating tasks that result in physical jobs being carried out, c) monitoring and analysing the fulfilment of tasks to measure the operational performance of the cell.

Generally speaking the inclusion of computer-based equipment with manufacturing cells has been limited to cells that deployed computer controls with a limited level of sophistication, including capabilities for example in respect of time and cost and utilisation of the parts and resources, and also in simulation and analysis of the system [50]. Indeed, according to Xiang [41] an unmanned cell has not been found to be cost-effective solution for a majority of manufacturing systems. Therefore it is probably that in most manufacturing cells in use today that the majority of tasks are carried out manually, especially in materials handling application areas. It may also be the case that the use of computer support for manual tasks is most commonplace in respect to tasks of types (a) and (c) above. Typically cells need to inter-operate with other (often supportive) manufacturing systems such as production planner, master scheduler, MRP (and ERP) software system, inspection, monitoring systems and so on. Hence the provision of computer support capabilities for tasks of type (a) and (b) may be achieved by extending the use of higher level software systems across cell, workstation and resource unit levels.
Important benefits that can be expected from arranging production operations into manufacturing cells are reported to include: lower set-up and throughput times; higher and more consistent quality per unit cost; improved material handling and utilisation; and an ability to reduce and simplify associated tools and equipment [38, 41, 42].

Definitions – Within the current literature, the term “manufacturing cell” is often interpreted differently. However, to understand and communicate concepts about a cell it is important to explicitly specify what a cell is. Different researchers have given various definitions for manufacturing cells, including:

Franks [51]: "A group of manufacturing resources consisting of machines and workstations which are organised and scheduled as an entity to accept discrete parts, sub-assemblies, and material, the cell then adds value through processing to create a new identifiable product as its output. Cell may be automated, semi-automated, manually operated or a combination of all three types."

Williams [52]: "Practical building blocks of CAM and CIM systems and are important as island of automation. Such cells usually consist of a number of closely co-operating different machines co-ordinated and controlled by a supervisory computer. "And also "A biological analogy as the smallest autonomous unit capable of sustained production."

On reviewing various definitions of manufacturing cells (also see appendix G for more definitions) given by different researchers, it was concluded that many existing definitions emphasis the ability of cell to flexibly produce a family of similar products. Many definitions also emphasise the important role of materials handling systems and control systems in a cell [38]. A need to develop and deploy a suitable integration and communication system between cell components and between the cell and other parts of an enterprise has also been usually considered to be an issue of concern [38, 51]. Despite occasional references to a need to resolve autonomy and automation issues in cells and the role of human supervisors, these issues have not been explored in any great depth in the manufacturing literature.

Assumption – Based on understanding gained from practical experience, this study will consider a manufacturing cell to be a segment of a larger manufacturing system (e.g. a shop-floor or a factory) that normally will be located in a confined and fixed area of a plant. Furthermore it will be assumed that manufacturing cells will have the following list of attributes:

- Generally they are designed to be adaptable with respect to the processing of a similar class of products or operations.
- They are constructed from human and technical resources that include: devices (e.g. machines tools); people (e.g. operators, supervisors); and control systems to assess performance of cells.
- Manufacturing cells may be designed to exhibit different degrees of automation and autonomy that match the nature of tasks performed in the cell, the availability of suitable human and technical resources and the needs of other systems used within the host enterprise.
- A control system (that usually will be computerised but may be only semi-automated) will be
deployed to (1) co-ordinate and control the flow of tasks within a cell (with reference to external supportive systems), (2) to organise and control the actions and interactions between control system and resource elements within and external to a cell (including external supportive systems) and (3) to monitor, analyse and report on the operation of the cell to external supportive systems.

- The resources in a cell may be classified into sub-groups to assist in the breaking down of a task within a cell and thereby to achieve better control over the cell. These sub-groups may have similar integration problems to that of a cell but on a smaller scale.

- Normally cells are semi-automated. Therefore human resources will normally be required to carry out certain operations and supervising tasks.

- A manufacturing cell is a segment of a manufacturing shop-floor. A shop-floor, in turn is a part of a manufacturing factory that will deploy other manufacturing systems. Some of these systems may have a product or function (type of operation) orientation (such as a cylindrical parts cell, an assembly shop, kitting cell or a test cell) whereas others may have a wider scope (e.g. a material handling system or an inventory system) or provide a fairly general infrastructure service (e.g. a quality or costing system). The cell control system will need to interact in an organised manner with these systems.

- The resources within cell may be dedicated to a cell or be reassigned without rearranging their physical location.

- The concept of cellular organisation can be applicable in other enterprise subsystems as well as for production subsystems. For example, team-based organisation groups of people working in technical-office environments can exhibit similar properties to a manufacturing cell.

2.1.2) Cell Controllers

It is necessary to structure and co-ordinate operations carried out by various elements of a manufacturing cell. Hence some form of control system is required to manage tasks performed within the cell and to allocate appropriate resources to specific jobs at the correct time. These control system functions are commonly realised by people with appropriate expertise and skills [53]. However their work can be supported by computer tools that enable them to fulfil their role more effectively, rapidly and consistently. Exceptionally it may be appropriate to completely replace people in this role by computers [41, 52].

A number of definitions of manufacturing cell controllers can also be found in the literature that consider control systems from different point of view [40, 41, 52]. However a fairly comprehensive definition was provided by Franks [51] follows:

"A manual and automated process, or both of them that accepts orders for work into the manufacturing cell and optimises the use of resources and materials to produce the required output: monitor and report on the progress of the requested work and the status of the manufacturing resources within the cell."
It is reported that a need to optimise the operations performed by cell entities and to monitor the status of cell entities have been key items to consider when designing various cell controllers [38, 43]. Normally such a capability can be provided by using a computer system to support the role of and interact with a human supervisor [42]. The optimisation processes should result in: an efficient and cost-effective use of resources; suitable scheduling and dispatching of orders; synchronisation and control of materials handling operations; and the handling of failures and error situations [43]. Monitoring processes should cover report generation and analysis of the status of physical devices (e.g. machine, tools, equipment), human resources and orders (e.g. jobs, due-dates, scrap rates, tolerances) [40].

To enable cell controllers to provide such capabilities their designer should have knowledge of (or access to knowledge of) the importance of common requirements that include the following:

**Configurability** - This property is important when designing a manufacturing cell to produce a wide range of products. Here a cell may need to be configurable in terms of being able to accommodate changes in product and task descriptions, and in regard to the availability of resource elements (including both hardware and software resources) [15]. To achieve the property of configurability in a cell control system the use of a modular reusable system design approach is recommended [28, 30].

**Connectivity** - Normally a cell controller will need to be integrated with higher level systems (such as shop-floor controller) as well as being connected to lower level device control systems [38, 43]. This should ensure that the overall operations carried out by a cell system are aligned to higher level goals in the manufacturing organisation. This emphasises the need for a consistent approach to integrating manufacturing systems.

**Portability** - The software applications used to realise cell controller should be portable to enable their operation in different computer systems, supported by various hardware platforms [38]. Portability is especially important when a cell controller functions in a multi-platform computer system, which is typically the case in the majority of manufacturing environments in which they are used [38]. Achieving this characteristic may well impact on the design and operation of more general integration schemes (that achieve connectivity) within the manufacturing organisation. The issue of software portability is also of importance to vendors of cell controllers, as they will wish to upgrade their products and support their use in many different environments.

**Scalability** - This is required to support various levels of technical complexity, such as complexity of the product specifications, the product quantity and its consequent difficulties regarding logistical complications [39, 40]. Scalability of a cell controller application may also be referred to

---

1 Although ‘portability’ is an important characteristic of the software tool used as cell control application, in this research for the sake of manageability or the research project the portability of the resultant application tools has not been the major concern.
the system domain that the application is usable.

**Sharability** - The cell resources (e.g. physical and information resources) need to be shared with other parts of the system to increase the efficiency of the overall cell system. This characteristic typically addresses the needs for integration, reusability, and generality of the system elements (modelling and physical systems) [35]. A large part of the manufacturing reengineering process is in fact similar and common to every type of the business process thus they could be captured standardised and re-used instead of developing again from scratch [19].

There are also other key metrics that need to be considered to improve the level of the control, such as: 'ease of use' e.g. by all operators and cell supervisors; 'maintainability' e.g. in the event of system failures; and so forth. In this research the centre of attention is on ensuring that cell controllers achieve improved configurability, connectivity, and sharability (e.g. reusability). Thus the following sections discuss some of the systematic approaches reported in the literature that seek to formalise the design of manufacturing systems, including the cell controllers.

### 2.2) Enterprise Integration

Enterprise integration concepts have been developed to formalise the design and specification of manufacturing systems. A key focus of these concepts is on connectivity amongst enterprise systems (including physical and modelled systems). This connectivity is typically expressed in terms of information, control, and material flows across the enterprise organisation [54]. According to Vernadat [19] within a manufacturing enterprise, a system integration should encompass different domains such as: market integration for consuming regional products; integration between manufacturing sites (or remote sites) with development organisations; integration between suppliers and manufacturers; integration between design and manufacturing; and integration between vendors of hardware and software components in order to improve interoperability and standardisation.

On examining integration issues within an enterprise, three major types of integration may be considered including system, application, and business integration [19, 55]. According to the CIMOSA consortium [56] system integration concerns physical connectivity between system resources so that they can actually exchange the objects. Application integration is concerned with sustaining the system portability in terms of information sharability, commonality of services, and interface unification. Business integration mainly focuses on harmonising decision making strategies within the enterprise and also with respect to relationships with other enterprises.

Enterprise integration should support the flow of objects (e.g. data, material, drawings, and tools). This includes the flow of management objects (e.g. decisions, targets, and finance). Relationships between suppliers, engineering, production and the customers to maintain the flow of technical
objects has been termed 'Horizontal Integration' (because the focus of concern here is on similar levels in the manufacturing hierarchical organisation). Realising connectivity between different system management level and functions and different operational level and functions, and the maintenance of decision flow has been refereed as 'Vertical Integration' [57](decisions are typically enforced from top levels of the hierarchy). On considering integration issue during the design of a system in a top-down manner, usually designers will encounter a massive complexity and design restrictions (and possibly high cost). This may be especially true in respect of a traditional hierarchical manufacturing structure with undefined direct relationships between the entities[43, 57] . When adopting a bottom-up design approach, each part of a system may be analysed in detail separately. However a lack of unification of the methods and standards used in respect of each part will not lead to integration on an enterprise-wide scale.

Hard-wired physical connections between system entities cannot normally lead to configurable, portable, scalable and widely connectable systems in manufacturing enterprises. Rather, flexible or so-called 'Soft Integration' of system entities [57] is required to enable systems to be adapted to changing requirements and conditions. In a soft integrated system the method of connecting entities should be clearly separated from the entities themselves. However this requires common, idealised models of system entities to enable their integrated operation via various integration mechanisms and techniques [19]. The use of soft integration approaches (based for example on the use of computational infrastructures like the use of Internet and CORBA [58]) can provide suitable ways of integrating multi-vendor supplied enterprise systems but on their own these concepts are not sufficient to realise enterprise wide integration' [57].

Two distinctly different approaches to solving the integration problems have been reported, namely 'localised' integration and 'infrastructure' integration [19]. A local integration solution achieves limited connectivity among a set of system entities. This approach to integration can be achieved using bespoke tools, possibly with some hard-wired intermediate interfaces, to produce a turnkey solution that proposes a unified vendor view or standard describing the whole system [36]. Whereas an infrastructure integration approach is based on use of an underlying communication method that underpins a defined hardware and software environment. The communication method facilitates various types of interaction and data exchange between sets of entities in the system. The entities of the system need to be prepared so that they can communicate with each other by using the services provided by the infrastructure, where the services will mainly be based on well-known techniques [29].

Generally, some approach to integration is needed to convey enterprise knowledge and deliver the

---

2 Here the term enterprise addresses a manufacturing company including manufacturing units and other supportive departments.

3 Arguably, a truly integrated system is likely achievable only for dedicated equipment in a system, supplied by a single vendor or a system that has been built completely based on unified and standard rules.
right information, to the right place at the right time [54]. By using modelling methods to implement the integration approach in an organised way it becomes possible to control the flow of information and knowledge within an enterprise and to formalise enterprise operations in a flexible manner. Information technology (IT) mechanisms can underpin the development of an integration and modelling approach to facilitate the development of the information services required to achieve enterprise integration.

2.3) Enterprise Modelling

An enterprise can be considered to consist of a number of processes that are realised concurrently by enterprise resources that contribute towards the overall objectives of the organisation [59, 60]. Enterprise models can be used to formalise the flow of control of those processes within an enterprise [39, 43].

The term ‘model’ has been defined differently in various fields of science and engineering. For example Barkmeyer [49] states:

"A model is an approximate design of some actual process or object. Engineers use models as a tool in designing and building physical objects. Similarly, system and software engineers use models to represent the characteristics of a system from several clearly points of view."

and Weston [57] considers a model as:

"a representation of some aspect of product realisation which can be used to facilitate visualisation, analysis, design, etc."

A number of definitions of the term used in different engineering fields were reviewed as part of this study (see appendix G – §G.2). A consensus of opinions on the definition of the term ‘model’ (that can be usefully deployed in the remainder of this thesis) can be expressed as:

"A model is a logical method to visualize the real status of an event (or process) in a system that can facilitate analysis and control of the system."

When seeking to formally characterise the properties of a manufacturing enterprise, modelling must be carried out from various foci of concern [61] (also see appendix G). Therefore a modelling strategy should be specified [19]. It should be made clear “why, what and how” a system should be modelled, what benefit can be expected from the model, and what degree of detail the model should contain. The degree of detail chosen should reflect the generality of the model and the “modelling level” or life-phase of an enterprise that the model needs to represent [62]. The generality issue concerns the scope of the domain in which a model applies (e.g. part of an enterprise, type of enterprise, etc.). Whereas the modelling level refers to the portion of the life-cycle formalised by the model (e.g. requirements capture, conceptual design, etc.). Furthermore, enterprise models capture certain perspectives (or foci of concern) about an enterprise, such as financial, business, information and function views. When formally modelling complex systems it is necessary to decompose (or breakdown) the system into manageable system elements. The modelling elements should preferably be defined in a generic and reusable manner to improve the
generality of the model (i.e. enable its use and reuse) in different domains and also to reduce the design time and cost of producing and using models [18].

There are many potential benefits from using enterprise modelling in respect of the life cycle of a manufacturing system [36, 63]. A model provides insights into system capabilities and highlights alternative solutions and application scenarios that prepare the system to adapt to business change[9]. Business change may influence many facets of an enterprise, including its processes, communication systems and information requirements, and the way that its resources are organised and operate [29, 64]. To satisfy new business or environmental needs a deep understanding of cause and effect relationships and constraints on change is required. Modelling methods can help to analyse alternatives and, to determine new system configurations that best fulfil requirements change before any real system reconfiguration needs to be activated [31].

In addition, modelling can help to establish appropriate integration methods [19]. It can enable decision-makers (e.g. analysts, managers, and consultants) to access conditions of the system and make better estimates about the future based on improved information about real capabilities. Potentially therefore the use of enterprise modelling can: reduce overall system lead-times and costs; decrease the inherent complexity of the system through use of improved organisational structures; promote improved match between design requirements and manufacturing capabilities; enable rapid changes; and improve system performance.

In the future the adoption of common modelling methods (by end-user manufacturers and by suppliers) should lead system developers to apply standard (or well-established) techniques during system design and configuration. Potentially this will improve opportunities to use off-the-shelf tools that are well proven and cost effective [6]. It should also enable the modelling of system components so that vendors can improve their capabilities in a more targeted way, leading to better and easier-to-use software and machine components [29].

When beginning an exercise of modelling a manufacturing system, various modelling design principles should be considered [65]. These may include:

- a modelling ‘strategy’ should be define to provide sufficient respond to the questions such as ‘what’, ‘when’, ‘how’, and ‘why’ the modelling can be beneficial,
- a modelling ‘methodology’ should be deployed to specify the main rules to govern the modelling process,
- a modelling ‘framework’ should define the links between modelling elements,
- a modelling ‘architecture’ should specify the relationships amongst the system elements, including models, resources, information, etc.,
- modelling ‘languages’ and presentational methods are also required to formally describe,
deliver, and present the modelling elements and constructs that will be used as elemental building blocks of a model,

- modelling 'view' should be specified to restrict the domain n of modelling coverage,
- also a modelling 'symbology' may be used to present models in a graphic representation.

In addition, if they are of suitable design and construction, enterprise models should enable:

- verification of the suitability of modelling elements,
- simulation based on one way of operationalizing models,
- rapid and effective reconfiguration when business processes change,
- reusability of the model and physical systems to improve their configurability and scalability,
- generality to cover existing models and the models to be built [66].

A comprehensive discussion on model specifications and requirements in engineering systems can be found in Appendix G, section G.2.

2.4) Information Paradigm

Information technologies and especially information modelling and information systems are often of paramount importance during the development of enterprise integration systems [67, 68]. An information system is used to design and develop the structures and procedures, which will make information available to system elements [69]. An information model is used to formalise those structures in a logical format.

Within the enterprise integration field, it is widely understood that information systems are required to possess certain characteristics including: they should embed object-oriented design concepts to facilitate system reusability and expandability; they should deploy standard integration methods and mechanisms to improve their connectivity; they should facilitate information sharing the information within the enterprise, reducing data redundancy and improving the operating efficiency of the system [66].

Use of a suitable information model can help to co-ordinate the flow of information and data within an enterprise [54]. A data model should allow the description of an information structure relevant to a system to be utilised in an implementation-independent format. Data modelling methods have generally been derived as aids to database design. Information modelling is related to the identification, representation and composition of data, information and knowledge that describes real objects [70]. There are basically two differences between data and information modelling. Data modelling is targeted at generating a model of data that is computer processable. Whereas information modelling is not, but could be computer processed. Information models need to be made explicit and formally documented. Therefore data modelling techniques can be used to
2.5) Enterprise Engineering Approaches - State-of-the-art

There have been many reviews of and comparisons made regarding enterprise engineering concepts and tools including those by Kosanke [72], Vernadat [19], Bernus [73], and Williams [74]. Examples of public domain enterprise engineering approaches are considered briefly in the following discussion.

**GRAI/GIM** – The Graphs with Results and Activities Interrelated/GRAI Integrated Methodology was developed at the University of Bordeaux in France [75] to help designers to model production management systems [76]. Initially GRAI/GIM focused on modelling decisional structures of a manufacturing enterprise related to strategic, tactical and operational planning. GRAI concepts were extended to support the design of CIM systems leading to GIM as an integrated methodology for business process modelling [72]. The scope of the GRAI/GIM framework covers conceptual, organisational, and physical modelling levels. The conceptual model is based on the application of ‘general systems theory’ [19]. This categorises any systems into physical, operational, decision, and information system elements [75]. This framework is intended to support the complete life cycle of manufacturing systems by generating a two-part specification of user-oriented and technically oriented issues [76]. GRAI/GIM provides a modelling capability that supports concept (analysis), structure (user oriented design) and realisation (technical oriented design) phases of the life cycle, with special emphasis on decisional aspects [72]. However, the GRAI/GIM framework does not include formal means of describing the implementation and behaviour of manufacturing systems [35]. For further reading about this approach the reader can refer to [75].

**ARIS** – ARchitecture for Information Systems was developed at the University of Saarbrücken in Germany [77, 78]. The ARIS approach focuses on issues related to enterprise information system design. Therefore it provides specific modelling support (i.e. IT concept support) for Information Technology (IT) parts of enterprise engineering projects [72]. ARIS is designed to target business process modelling at database design. It achieves this by providing modelling constructs that lead to the development of entity relationship data-models [35].

The ARIS approach to modelling has some similarities with CIMOSA as both have a main focus on traditional business-oriented issues such as order processing and production planning [78]. ARIS supports a number of modelling perspectives (including function, information and resource views). It also enables system designers to develop a ‘Control View’ modelling perspective as a practical and flexible means to manage and control the integrated use of other modelling views during system implementation and execution [35, 79]. However, the flexibility of the ARIS approach can lead to lack of solidarity and increased complexity in the resultant data-model. This can cause major difficulties when designing applications and supporting data...
models on an enterprise-wide basis [80]. The ARIS approach has a scope that supports enterprise modelling from the development of operation concepts, through IT concept development to IT system implementation [72]. Therefore the generality characteristic and conceptual formalisation methods have not been the centre of attention in this approach [35]. For further reading on the ARIS approach refer to [78].

**PERA** – The Purdue Enterprise Reference Architecture was developed at the University of Purdue in USA [81]. The PERA methodology is characterised by its layered structure. The life-cycle starts with a definition of the Business Entity to be modelled, identifying its mission, vision, management philosophy, mandates, defines project sponsors, leaders and members, etc. and ends with obsolescence of the plant at the end of the operational phase [72], with implementing a pseudo time scale [82]. PERA concepts have a scope that covers the introduction, implementation and operation of an enterprise business entity. These entities may be either part of a larger entity or be the complete enterprise itself. PERA can support and guide the development of the Master Plan for an Enterprise Business Entity [72]. PERA does not provide its own modelling methods. However it can be supported by other modelling tools and techniques. Consequently this approach exhibits certain difficulties, as it cannot be realised and supported by computer-processable methods in a well-defined way [81]. The PERA framework places a particular emphasis on human factor issues in enterprise engineering projects [35]. Further literature on PERA can be found in [82, 83].

**IEM** – The Integrated Enterprise Modelling approach was initially developed by the Fraunhofer Institute in Germany [60]. IEM supports the creation of enterprise models for business re-engineering. It supports the modelling of process dynamics to enable the evaluation of operational alternatives [72]. IEM concepts have a scope that covers the main phases in the life cycle of enterprise engineering projects, including requirements, design, implementation and model up-date. The IEM approach employs the SADT/IDEFO activity box, CIMOSA behavioural rules and takes advantage of object-oriented approaches to software system design [84]. IEM concentrates on two modelling perspectives, namely information and function views. For further reading refer to [19].

**GERAM** – The Generalised Enterprise Reference Architecture and Methodologies (enterprise modelling framework) has been defined by the IFAC/IFIP Task to provide necessary guidance for enterprise engineering processes. The workforce has sought to develop the GERAM specification as a semantic unification of concepts and models used in public domain enterprise engineering approaches [54]. Therefore GERAM has been designed as a reference model of engineering architectures and methodologies [72]. The structure of the GERAM architecture actually encompasses a selected combination of concepts from the CIMOSA, PERA, GRAI/GIM approaches [74]. The so called general concepts identified and defined by GERAM
cover enterprise engineering project life-cycle, life history, and model views [54]. GERAM identifies and proposes a set of components for use in enterprise engineering [85]. These components include: Generalised Enterprise Reference Architecture (GERA); Generic Enterprise Engineering Methodology (GEEM); Generic Enterprise Modelling Languages (GEML); Generic Enterprise Modelling Tools (GEMT), Generic Enterprise Models (GEM); Generic Enterprise Modules (GM); and Generic Enterprise Theories (GT). For further discussion on the GERAM architecture see appendix G and [62].

CIMOSA – CIM Open Systems Architecture has progressively been developed by the AMICE Consortium [56] within a number of ESPRIT Projects. CIMOSA was designed to help companies to manage change and thereby to integrate their facilities and processes to face world-wide competition [86]. The CIMOSA architecture supports process oriented modelling of different manufacturing enterprises. It also provides execution support for the operational phase of manufacturing systems [87]. The CIMOSA framework supports the engineering of enterprise models with a scope that covers requirement definition through to implementation description and the operational use and maintenance of manufacturing systems.

The CIMOSA modelling framework provides the user with architectural constructs and guidelines for the structured description of business requirements and their translation into system design and implementation [88], as illustrated by Figure 2.1.

The Derivation Process guides the user through the three modelling levels: from the definition of enterprise business requirements (Requirements Definition) through the optimisation and specification of the requirements (Design Specification) to the implementation (Implementation Description).

On each modelling level in the Generation Process the enterprise is analysed from different viewpoints (Modelling Views). CIMOSA defines four modelling views for different aspects of an enterprise, including:

- The Function View describes the workflow of the Enterprise Functions
- The Information View describes the inputs and outputs of the Enterprise

Figure 2.1: The CIMOSA modelling approach
Functions

- The Resource View describes the structure of resources (Humans, machines, Data Processing programs) required to perform the Enterprise Functions.

- The Organisation View defines authorities and responsibilities regarding functions, information and resources.

To reduce modelling effort, CIMOSA defines three levels of generality from purely generic to the highly particular. The first Generic Level is a reference catalogue of basic CIMOSA architectural constructs (building blocks) for components, constraints, rules, terms, service function and protocols. The second Partial Level contains a set of partial models applicable to a specific category of manufacturing enterprises. The third Particular Level is related to one particular enterprise and is defined in the Instantiation Process by the modeller using already prepared building blocks from the Generic and Partial Level and developing new particular enterprise-specific components.

The CIMOSA function model consists of a set of modelling constructs (or business entities) that decompose functional processes into structured modelling entities [89], as illustrated by Figure 2.2.

The business entities include the following modelling constructs:

- **Domain** is a construct, which is used to define the part of the enterprise relevant for achieving a defined set of business objectives, i.e., it is used to specify the overall scope and contents of the particular model of the enterprise. A Domain description consists of: Domain Objectives and Domain Constraints, Domain Relationships describing the Domain Boundaries, Domain Objects, and Domain Processes.

- **Domain Process** is a construct used to define which Enterprise Functions influence the achievement of the related Domain Objectives. The Domain Processes are identified during the establishment of the Domain. Each Domain Process is then expanded in terms of the generic Enterprise Function construct during the function decomposition phase.
• **Business Process** is a special type of Enterprise Function, which aggregates all the lower level **Business Processes** and/or **Enterprise Activities** required to carry out the defined tasks and defines the complete sequence of operation for these activities. A **Business Process** always has a functional, behaviour and structural part defined, and is initiated by an Enterprise Event so that its execution will result in the fulfilment of the identified business objectives.

• **Enterprise Activity** is a special type of Enterprise Function and is defined as a non-decomposable or low-level Enterprise Function. **Enterprise Activities** describe the basic functionality of the enterprise. **Enterprise Activities** are not part of any given **Business Process** as such, but are utilised by one or more **Business Processes** through their associated set of **Procedural Rules**. This relationship of **Enterprise Activities** and **Business Processes** via the **Procedural Rules** make it possible for the sharing of **Enterprise Activities** amongst different **Business Processes**, and also accommodates the behaviour changes of the enterprise by only altering the set of **Procedural Rules** while maintaining the basic functionality of the **Enterprise Activities** intact. At the design specification modelling level **enterprise activities** may be further decomposed into **Function Operation**, which can operationalize **enterprise activities**.

It must be emphasised that some level of understanding about CIMOSA modelling structure and methods are required to realise the modelling approach developed and discussed in this thesis. Therefore the readers are recommended to review a more complete description of the CIMOSA approach given in Appendix G (§G.4.1).

2.5.1) Modelling Methods and Languages

Many modelling methods and languages have been developed and used to underpin aspects of enterprise engineering projects. They typically provide formal modelling descriptions in a logical and a computer processable format. A selection of modelling methods and languages that have been used in respect to large scale enterprise engineering projects are outlined below.

**IDEF** - IDEF (Integrated Definition) methods provide standard modelling and analysis methods for enterprise engineering [90], They are designed to support common enterprise engineering activities such as discrete-event simulation, information system requirement assessments, and activity-based costing. For example, IDEF0 is a method designed to model the decisions, actions, and activities carried out by an organisation or system [91]. IDEF methods can be classified based on their applications. IDEF0 was derived from a well-established graphical language, the Structured Analysis and Design Technique (SADT). IDEF0 is useful in establishing the scope of an analysis, especially for a functional analysis [92]. IDEF1 was designed as a method for both the analysis and communication of information requirements [93]. IDEF1X is a method for designing relational databases with syntax designed to support the
use of semantic constructs necessary to develop a conceptual schema [94]. The IDEF3 Process Description Capture Method provides a mechanism for collecting and documenting processes [95]. IDEF4 is used to design systems centred on the use of object-oriented methods to decompose activities into the manageable parts [96]. The IDEF5 method provides a theoretical and empirical method designed to assist in creating, modifying, and maintaining ontologies [93]. For further reading on IDEF methods see [90, 97].

STEP - The Standard for the Exchange of Product Model Data (STEP) is an ISO standard (ISO 10303). STEP was developed to provide mechanisms for the representation and exchange of computerised product models in a neutral form, i.e. independent of any particular computer system [98]. Three sets of implementation issues have been addressed by this standard, namely: physical files exchange; application programming interface issues; and database implementations. Each implementation method is specified by mapping from the EXPRESS language [99] onto the formal language used for the method. Physical file exchange is the method of writing and reading of data (corresponding to application interpreted models) into a sequential file format referred to as a STEP physical file format [100]. Fundamental principles on which the STEP standard is based include: separation of product information from implementation methods used for data exchange; common representation of product information to many applications; using formal data specification languages (EXPRESS) to specify the product information; specifying implementation methods that support the representation of product information, included in application protocols; and providing a framework for conformance testing of implementations [101].

EXPRESS - EXPRESS is a formal language [102] that was chosen by the STEP community as a means of describing product data. EXPRESS is human-readable (by an expert) and is readily computer-processable. An EXPRESS program defines schemata comprising 'Entities'. Entities are identified by their 'Attributes' and relationships with other Entities. The 'super-type/sub-type' construct defines inheritance relationships between a child Entity and its parent Entity. A graphical representation technique is also available to aid in understanding and facilitating use of the language and can be used to represent data in an understandable manner. This aids human communication. EXPRESS-G [99] was developed as a formal graphical notation to aid human interpretation of data subsets in EXPRESS. It provides constructs to define; schema and inter-schema links; entities; attributes; relationships; types; and multi-page referencing.

Petri-Nets - Petri-Nets (PN) formalisms can be used to underpin modelling methods and languages. They can facilitate visualisation and analysis of the behaviour of complex and concurrent processes in a system, where the processes share resources [103]. The basic Petri-Nets formalisms have been extended in different ways by various research groups leading to Timed PN [104], Stochastic PN [105], Coloured PN [106]. Petri-Nets formalisms have been
used to assist in the visualisation, analysis and control of enterprise approaches systems. For example, enterprise modelling descriptions developed as part of CIMOSA and IDEF3 approaches can be readily converted into the form of stochastic Petri-Nets models [105]. The reader can refer to [103] for further information on Petri-Nets formalisms.

**KIF** – The Knowledge Interchange Format [107] can be used as the basis of a computer-oriented language to formally specify the interchange of knowledge among disparate programs. It has declarative semantics (i.e. the meaning of expressions in the representation can be understood without recourse to an interpreter for manipulating those expressions); it is logically comprehensive (i.e. it provides for the expression of arbitrary sentences based on the use of first-order predicate calculus); it provides constructs to represent knowledge about the representation of knowledge; and it facilitates the definition of objects, functions, and relations [108].

**CORBA** - The Common Object Request Broker Architecture [109] was developed by the Object Management Group (OMG) [58]. The CORBA was designed to distribute applications across client-server networks. The goal of CORBA is to make programming easier by ensuring that CORBA-based applications are highly portable. Developers of CORBA applications can have a standardised set of facilities [110] to communicate and establish interaction between elements of a system. The paradigm CORBA follows is a combination of two previously existing paradigms. The first of these is distributed client-server computing. In this respect CORBA is based in part on message-passing systems, most commonly found in UNIX-based environments. The second paradigm deployed by CORBA systems is that of object-orientation [110]. The OMG has been successful in encouraging vendors to agree on definitions of standard objects and methods.

**KQML** - The Knowledge Query and Manipulation Language [112] is a language and protocol for exchanging information and knowledge. The development of KQML has been part of a larger effort, the ARPA Knowledge Sharing Effort. KQML is both a message format and a message-handling protocol designed to support run-time knowledge sharing among software agents. KQML can be used as a language to develop application programs that interact within an intelligent system or to enable two or more intelligent systems to share knowledge in support of co-operative problem solving [112].

**ALBERT** - The Albert language is a formal agent-based language designed to enable the formal definition of architecture and behaviour of systems [113]. It is claimed that this language can fully formalise modelling descriptions that have been partially formalised by the CIMOSA

----

4 However, Microsoft has decided to create its own Object Request Broker system called Common Object Management (COM) [111] within its Object Linking and Embedding (OLE) technology.
modelling structure at the generic level. Further information on use of ALBERT language can be found in [114].

**SQL** - Structured Query Language is a language used to create, maintain and query relational databases. It is an ISO (ISO/IEC 9075-1992) and ANSI (ANSI X3.135-1992) standard. SQL uses regular English words for many of its commands. This makes it relatively easy to use. It is often embedded within other programming languages [115]. Many database products support the use and manage of SQL queries and provide proprietary extensions to the standard language. In 1993, the ANSI and ISO development committees decided to split future SQL development into a multi-part standard. The Parts include Framework, Foundation, SQL/CLI, SQL/PSM, SQL/Bindings, SQL/XA, SQL/Temporal [115].

**2.5.2) Modelling Tools and Techniques**

There have been many modelling and system engineering tools developed by both research and commercial sectors to support enterprise engineering projects. Some of these tools that were understood to be directly related to this study, are discussed below.

**ARIS Toolset** - The ARIS Toolset is part of the ARIS Product Suite (by IDS Scheer, Inc [78]). The toolset comprises a range of tools for business process improvement, ERP implementation and optimisation, knowledge management, benchmarking, simulation, activity-based cost management and workflow analysis. The ARIS Toolset (version 4.0) provides a powerful tool that can be used to graphically display and optimise organisational processes and operations. This tool has what the vendor claims is a component-based architecture made up of four main components: Designer, Tables, Wizards and Explorer. This provides a platform that facilitates the reuse of objects and common use of models. The ARIS Toolset Version 4.0 also supports mapping to systems descriptions developed using the Unified Modelling Language (UML) [116].

**CIMBIOSYS** – Computer Integrated Manufacturing - Building Integrated Open Systems provides integration framework to run applications developed using a modelling environment. CIMBIOSYS was developed by a number of researchers at the MSI Research Institute in Loughborough University. The design and application of CIMBIOSYS has had a major influence on system integration projects carried out in this research institute (including this research). CIMBIOSYS is described further in appendix G (§G.4.3).

**SEWOSA** – The SEWOSA CASE Tool [117] was also developed by researchers at the MSI Research Institute at Loughborough University as part of the EPSRC funded Model-Driven CIM project [118]. The SEWOSA tool implements an enterprise engineering approach that builds on and extends CIMOSA concepts. It complements the use of CIMOSA concepts using Petri-Nets simulation and modelling capabilities, object-oriented design tools and the services
of the CIM-BIOSYS integrating infrastructure services. SEWOSA has been designed with a scope that covers the modelling of manufacturing systems from requirement capture through conceptual design and detailed design (i.e. model building) to system execution and change by enacting models to facilitate simulation, and rapid-prototyping. Importantly the runtime operation of systems (designed and constructed by SEWOSA) is model driven. In this way the configuration of systems built using SEWOSA can be altered in an organised and rapid way by changing the model that drives the runtime operation of a system. Further discussion on SEWOSA is included in appendixes E and G (§G.4.4).

McCIM - This is an integrated set of tools based on MAP 3.0 [119]. The McCIM toolset was designed to demonstrate the application of CIMOSA concepts. It was claimed to be the first CIMOSA prototype tool. McCIM was developed within the ESPRIT project VOICE (Validating OSA in Industrial CIM Environments) primarily by Institute of Applied Informatics (Nuclear Research Centre Karlsruhe). The main objectives of developing McCIM were: to demonstrate the application of the CIMOSA concept, to allow the capture and validation of CIMOSA models and services, to provide means of rapidly prototyping CIMOSA applications and services and to provide a training and education tool to promote understanding of CIMOSA [120].

CIM-Tool – This is a commercial application tool developed by RGCP (René Gaches Consultant in Production). RGCP is an independent Consultant, who works for large European companies, in the field Computer Integrated Manufacturing [121]. CIM-Tool provides graphical representation of model descriptions developed based on an approach called 'CimOsa/rg' [121] that is derived from the CIMOSA modelling architecture. CIM-Tool provides a capability to capture information and functional activity models. It supports interfaces to other commercial applications that use standard interaction methods to access and update data storage systems.

IThink – IThink is business process analysis tool developed and sold by the company High Performance Systems [122] in the USA. Model building with IThink is achieved via three parts, namely: creating a flowchart-like map of the process or pattern of influences that need to be modelled; configuring and tuning the model; running the model, and viewing and analysing its output via graphs and tables. The layering capability of IThink enables the users to preview a specific representation of the model, such as manufacturing process, information model, financial model, and so on. The IThink tools incorporate a dynamic simulation capability that can be used to analyse and design system processes [123].

2.5.3) Standards

Standardisation efforts are needed to improve the understandability of enterprise models and to facilitate interoperation between models created by different partners [72]. Effective standards can be very helpful for manufacturing end-users of models as well as for providers of IT systems. Some
important ongoing standardisation efforts related to enterprise engineering include:

- **EMEIS**: The European standards effort on Enterprise Model Execution and Integration Services aimed at generating standards for infrastructures [124].

### 2.6) Component-Based Approaches

In recent years so called 'component-based technology' has emerged and begun to be used in the commercial marketplace [125]. Manufacturing industries have for many decades appreciated the benefits of moving from custom development to assembly from pre-fabricated components [33]. Component-based manufacturing can deliver many products to the marketplace that would otherwise be prohibitively expensive.

The benefits of modular equipment and system design and build have been appreciated for many years. It follows that it is a natural development to design and construct manufacturing systems based on a component-oriented approach. Indeed various types of system component have been referred to as system building blocks [85], reusable objects [126], system Holons [48], and so forth. If a traditional hierarchical functional decomposition approach to design is followed, the lowest level of a function hierarchy will be so called 'atomic activity' building blocks. If these blocks can be well defined they may be reused as components.

An enterprise may be distributed around the globe and comprise heterogeneous business units, possibly functioning as a virtual enterprise [23]. One or more of these business units could themselves comprise 'Holonic Manufacturing' building blocks that function as holons [48]. Three main types of 'holon' (i.e. autonomous building blocks) have been defined (namely orders, products and resources) that can themselves be decomposed into reusable components, that structure functional 'holons' [127, 128].
The use of component-based technology has been considered in respect of different aspects of enterprise engineering including the functional models (or modules), information models and resource models (or modules) [129], where formal descriptions of such modules (or components) have been used, mainly during system design and implementation. The development of concepts related to component-based manufacturing systems has been influenced greatly by the trends in component-based software engineering [126, 129]. This emerging discipline has already developed means to capture, structure and generate software components.

The fundamental idea behind the use of software components is based on the 'buy, don't build' philosophy advocated by Fred Brooks [130]. Whilst the design of software components is founded largely on the notion that certain parts of large software systems reappear regularly [126], these common software parts can in many cases be written once, but reused several times. Indeed in software system design arguably the use of software components is a natural extension of ideas related to the use of subroutines and procedures.

Component-based system development concentrates particularly on the reusability of system components. However in practice it may be necessary to spend a significant amount of reengineering effort on developing these components before they can be adapted for use in new systems [131]. Moreover, adapting pre-existing components for use in a new system requires integration techniques such as the use of an 'Application Programming Interface' [132], wrapping, bridging, or mediating, as well as developing an increased understanding of architectural interaction issues and component properties [132].

The application of component-based techniques can potentially facilitate the design of change-capable manufacturing systems in respect to configurability and affordability, by reducing the cost and time involved, by building from pre-built, ready-tested components, and adding value and differentiation by customisation for specific customers [28].

The deployment of the component-based approach will be discussed in greater detail later in this document (§6.4).

2.7) Review and Discussion
Organisational levels in manufacturing enterprise have been classified into a hierarchy of levels, namely: factory, shop floor, cell, workstation, and device levels of operation (§2.1). This research is focused on the cell level and seeks to improve design and reconfiguration methods used to produce and change manufacturing systems at this level. A manufacturing cell is considered to be a group of manufacturing devices and computer tools. Therefore typically in a cell both human and technical resources need to interoperate to produce a specific class of product by carrying out a specific set of manufacturing operations (or activities) in a semi-automated manner. However, the cell and its constituent elements must also interoperate with other factory systems to realise business-oriented requirements (§2.1.1).
As mentioned in section 2.1.1, manufacturing cells typically perform three main types of task, namely: receiving jobs from the shop-floor level and allocating them as sets of related sub-tasks to cell resources; co-ordinating the interoperation of resources to achieve jobs; monitoring the condition of cell elements and their progress towards task completion.

Therefore cell control systems are required to plan, synchronise, control and monitor performance within the domain of the cell to ensure the timely fulfilment of jobs assigned to the cell. In general a cell control system should be capable of reconfiguration so that it can respond to or accommodate changes in requirements or operating conditions. Other factors that need to be considered include the need for: a generic design structure to ensure expandability, connectivity and general applicability (i.e. reusability and portability); a consistent and flexible interaction method between cell elements and between the cell and other manufacturing systems, e.g. to improve the sharability of resources and thereby to improve the overall efficiency of the enterprise (§2.1.2).

The use of an organised enterprise engineering approach should facilitate the design and construction of cell control systems so that they fit their environment well. Various approaches to enterprise modelling (§2.3) can be used to capture and formalise descriptions of cell systems. They can also form part of a modelling environment designed to facilitate the life-cycle engineering of cells. Ideally the scope of this environment should be broad to support requirements capture, cell system design and design analysis and cell operation and reconfiguration.

Various enterprise engineering concepts, methods and tools were reviewed (§2.5). The findings of a comparison between the capabilities offered by available enterprise modelling approaches, made by Kosanke [72] are summarised in Appendix G. Many of the modelling approaches reviewed provide support for key aspects of enterprise engineering projects. However none of them provides comprehensive life-cycle support nor are they geared towards the engineering manufacturing cells and their control systems via a component-oriented design and construction paradigm.

Within the context of this research, it was concluded that the life-cycle engineering approach required in this research should have the following characteristic properties:

• It should cover operational details as well as conceptual aspects of manufacturing cells;
• It should specify with sufficient completeness formal means of defining function and information aspects of cell systems in both generic (i.e. applicable in different cell domains) and specific (to a particular instance of a cell) forms and be expandable to cover other foci of concern about cell systems as required;
• It should have an 'open' structure to enable cell models to be transferred to and from and be used within other modelling tools;
• It should lead to the modular design of cell models and cell solutions to facilitate the adoption of the life-cycle engineering approach when dealing with various types of manufacturing cell, e.g. of different scope and complexity. Ideally the approach should be practical and feasible for
small or large cell systems.

After an appraisal of the capabilities of a number of modelling approaches, a decision was made to build upon use of the CIMOSA enterprise engineering approach. The rationale for this decision is summarised below:

- CIMOSA has a well-established modelling architecture that has been studied, tested and validated by many academic research groups and by industrial end-users in different manufacturing sectors (see Appendix G).

- Many parts of the CIMOSA architecture have been considered as the basis of CEN and ISO standards.

- The organised, largely top-down structure of CIMOSA was considered suitable for many manufacturing systems. Also the modular construction of the CIMOSA architecture should ensure the compatibility of this approach with other emerging paradigms such as object-oriented, distributed objects, component-based, and agent-based approaches. This potentially should facilitate reusability, sharability and scalability of resultant system designs and solutions.

- CIMOSA is well structured to support system specifications at a generic level, as well as at a particular level. It also includes some (albeit insufficient) formalisms to underpin system execution. A need for suitable structure and formal concepts was realised to be particularly important requirement for this research.

In addition, the CIMOSA architecture has been used successfully in other research projects carried out within the MSI Research Institute in Loughborough University. This has provided valuable relevant experiences and associated methods, and tools developed by researchers in the Institute. It was realised that this would provide help throughout this research.

However, a number of limitations have been observed and reported on about the use of the CIMOSA approach. The primary constraints include:

- CIMOSA models are mainly intended to provide operational support rather than act as project guides when developing or re-engineering business entities [72, 87],

- Weaknesses in respect to its concepts in support of human factors [35],

- Lack of sufficient concepts regarding high level business perspectives, including economic aspects of a system [133],

- Lack of methodological guides for modelling systems to generate specified solutions at operational levels [117],

- CIMOSA was designed originally for use in engineering discrete manufacturing systems and few comprehensive examples of its use in other domains have been reported [73].
The possible impact of these limitations was understood when the CIMOSA approach was selected for this research. However, it was expected that in the domain of this research the known limitations could be overcome by adopting the use of complementary enterprise engineering approaches. Therefore it was concluded that the research should not be overly restricted by CIMOSA shortcomings. Also ongoing standards work on GERAM continued to be influenced greatly by CIMOSA concepts. Hence it was expected that industry would wish increasingly to exploit CIMOSA and would need to know more about its applicability and actual constraints.

Regarding the modelling of information aspects of cell control systems, a decision was taken to adopt use of EXPRESS modelling language as part of the STEP standard methods. The main reasons for this selection were as follows:

- EXPRESS and STEP have been widely used within manufacturing industries,
- EXPRESS is an international standard data modelling language,
- EXPRESS has been tested and validated by academic research groups, including the MSI Research Institute,
- EXPRESS is being used by various IT vendors and there are many off-the-shelf applications which can facilitate the use of this language,
- Regarding specific requirements of this research, the EXPRESS language can readily capture information descriptions defined by CIMOSA and develop its own information model based on those descriptions.

In addition to selecting enterprise modelling approach and information modelling methods, it was assumed that a component-based paradigm would need to be selected or developed to facilitate key aspects of the life-cycle engineering of change capable manufacturing cells.

Before reading the remaining of this thesis, the author would like to remind the readers that some basic knowledge of CIMOSA modelling constructs (e.g. domain process, business process and enterprise activities) are essential to understand the modelling methods used in this study. Therefore the reader is strongly recommended to consider reading supporting literature in Appendix G.
Summary
This chapter has reviewed literature on the design of manufacturing cell control systems. It also reviewed various approaches to enterprise engineering and ways of formalising concepts needed in support of the life-cycle engineering of manufacturing cells. Background and emerging literature on component-based approaches was also considered. This literature review has provided the basic rationale for selecting an enterprise modelling and an information modelling approach. It has also developed general requirements and constraints related to the project’s aims, within the context of the design of manufacturing cells and the construction and reconfiguration of cell control systems from reusable components.
3.0) Introduction
The function of a cell controller can vary depending on the type, scope, and level of autonomy of decision making required and on properties of the operational environment of the cell. The major functions of the cell are to receive orders, schedule the cell resources, process the operations, and to monitor the status and results. It also needs to react to abnormal conditions in the cell. The cell capability should be checked before accepting new jobs. Shop-floor orders should be decomposed into individual jobs and be distributed between cell resources along with necessary engineering knowledge. The status of jobs and resources must be carefully maintained and analysed. Furthermore, cell controllers should also communicate with external systems and interact with users. Therefore, the control of a manufacturing cell presents complex problems and requires innovative effort to solve related design problems.

3.1) Manufacturing Cell Design and Control Problems
According to the result of a survey of companies developing manufacturing cells [42] the most common reasons for using manufacturing cells are to reduce: WIP inventory; set-up time; throughput time; material handling; cost; and finally to improve output quality.

This is achieved mainly by tackling problems within cells (inter-cell problems) as well as by considering out-of-cell communication issues with other systems in enterprise.

Some of the main problems reported by different literature are outlined below [38, 41, 42, 50, 51, 134, 135]:

a) Inter-cell problems
- Human resource related
  - Operator skills and team working
  - Operator efficiency and capability
  - Operator work balance
- Machine/Device related
  - Capacity and technical capability
  - Maintenance and break-down
- Process related
  - Volume/Load balance
  - Cell Scheduling
  - Flow of data/material/devices
- Control/Analysis related - Control the cell
entities and analyse the results and quality

b) Out-of-Cell problems

- Cell requirements
  - Hardware (e.g. material, tools)
  - Software (e.g. data, scheduled work, orders)
- Technical Capacity/Capability - reorganisation and control of cell availability, capability, and efficiency
- Connectivity - integration of cell elements with other parts of the system to maintain process consistency
- Communication protocol - communication techniques to improve connectivity and exchange information
- System design issues

Figure 3-1 provides an overview of some of the problems commonly encountered in the design and customisation of manufacturing cell control systems and indicates the area in which greatest emphasises has been placed during this research study.

It is probable that many of these problems arise because ineffective systems design and systems implementation methods and techniques are developed and used. It is also probable that key area in which improvement can be made concerns the co-ordination of resources so that they collectively meet overall targets of the system. If these assumptions were indeed true, then significant benefit might be achieved by developing systematic methods and computer tools to organise and support design and implementation activities involved in achieving (a) effective control and monitoring of cell entities during the operation of cells, and (b) improved cell design (in terms of analysing system requirements and capabilities) to ensure that sufficient flexibility is provided to configure the system resources in such a way that a given cell can accommodate planned changes, and even certain type of unplanned changes.

Essentially this study seeks to investigate these assumptions and to develop improved methods and tools. Therefore, this research has considered how the co-ordination of cell entities can be achieved in a flexible and changeable way to enable cells to realise various (pre-programmed) product specifics (e.g. colour, size), to introduce new products with a short lead-time, and to accommodate certain type of unexpected change.

In addition, it was necessary to study problems associated with contemporary means of constructing and operating manufacturing cells. Here a number of common problems were identified and generally applicable solutions proposed and used as part of a new approach to the design and construction of manufacturing cells.

Indeed, on considering current methods and techniques used to engineer cells two extreme strategies can be observed, namely: top-down and bottom-up approaches.
Naturally, a top-down approach will focus on the definition of requirements and generic modules of control that can be reused, i.e., it will attempt to define conceptual requirements and solutions in a way that is abstracted from implementation specific details [36]. The design of such systems is very complex but there may be very important system benefits arising from the use of a high-level conceptual design principle leading to configurability and adaptability. However, in some cases, processes and decisions need to be centralised. Indeed the system responses may be slow and/or the system may consume greater resources. In addition, such system design approach may not be fundamentally able to develop and use a structural design decomposition that can facilitate properties like reuse and reconfiguration. For instance, it may not be practical to separate out certain issues that have an inherent tight coupling [34, 49].

Typically, a bottom-up approach is focused on constructing the system based on a detailed knowledge of control modules and more local but complete specifications of the operation of a system [34, 36]. It may result in a shorter system engineering lead-time when developing a cell system for use in a fairly limited application domain. However, it will often result in a one-off solution, which cannot readily be reused e.g., in a developed or enhanced form or in a different application domain or manufacturing plant. Naturally, bottom-up design starts with a given decomposition of achievable component elements then focuses on integrating the elements into a whole system. Here a cell control system may be broken down into a set of functional elements (or modules) that collectively control and monitor cell resources in a certain way.

At present, for the general case, neither of these extreme approaches are sufficiently capable of developing cell control systems that exhibit sufficiently high level of performance, yet flexible, and suit the needs of both end user manufacturers and vendors (i.e., system application providers) [15, 36, 42].

Various means of modelling cell systems are reported in the literature that have the potential to address the problems and issues mentioned above [38, 42]. However, the isolated use of a modelling technique and a standard communication protocol will still in general lead to inflexible mappings between enterprise models (i.e., process and activity definitions, that describe what needs to be done by when) and physical elements (i.e., resources which realise the what and by when) [57]. Thereby, contemporary approaches may result in inflexible links between resource elements of manufacturing systems [21, 57]. Inherently such an outcome will constrain both the scope and reuse of a system.

In addition, a potential problem may arise because existing theoretical models of manufacturing cells reported in the literature essentially encapsulate only a single and fixed process thread, i.e., a process flow, required to manufacture a specified product via a given set of activities. This will not cause major difficulties if the model is applied in domains characterised by fixed processes and a fixed grouping of resources, such as might be found in a casting cell designed to produce a small
number of parts. However, the application of such a model in support of the design and construction of a flexible production line (requiring for example to manufacture of hundreds of different parts) the relevance and suitability of the model will be open to question.

A further concern is that many emerging approaches aimed at enabling cells to adapt to operational change, are based on the application of a large number of pre-defined rules for a specific domain. They do not consider the provision of flexible linkages between control components and resources available to a cell. Therefore, this research is founded on the premises that to date the industrial application of enterprise modelling methods has been limited and falls well short of its potential. On the other hand, modelling tools and supporting software application packages have been used successfully to formalise the design control and utilisation of resources in respect to fairly fixed processes. However, based on the use of existing enterprise engineering approaches if the process or resources used in a cell change, then a new model of the cell or production line has to be built. In effect under such conditions the modelling tools support use of no more than a set of predefined operations and therefore one might expect their use to have little advantage over conventional expert (or rule-based) systems.

With current approaches to designing and implementing manufacturing cells, constraints in system scope will typically arise as a result of either:

- unmanageable complexity as the scale of the system grows, e.g. giving rise to unacceptable cost and lead-times associated with changing a system,
- the use of inflexible links between resource and control system elements,
- inherent difficulties in achieving system interoperation across boundaries with other systems in the host environment; i.e. with systems previously (or in the future will be) installed into that environment which may well have different underlying concepts, technologies and origin (this class of limitation arises because of so called 'legacy system' problems) [23, 57].

In general, the need to: a) improve resource co-ordination during the operation of manufacturing systems, b) bridge the high-level abstract and formal system models and low level system operation (also see figure 1.3), c) realise flexible mappings between system entities (e.g. resources and supporting tools and techniques) was identified. Hence addressing these issues was selected as the subject of the research study with the aim of improving the status of manufacturing cell design and construction methods and tools.

1 To address this problem some research studies [38] have adopted a decomposition of generic cells into three main classes of system components, namely: the cell database; the knowledge base; and reasoning method. These approaches are mainly rule-based (e.g. deploy expert systems).
3.2) Research Approach – The Primary Vision

Based on previous practical experience and literature review, it is therefore assumed that present means of designing and constructing cells and cell control systems will constrain enterprise engineering projects. A manifestation of this will be difficulty in maintaining alignment between enterprise goals and the operation of cell systems. Furthermore, the use of contemporary approaches to system design and construction will result in an inability to respond to, or accommodate unforeseen change without long time delays and high-cost enterprise engineering projects.

Thus arguably significant benefit would occur for end user manufacturers, system developers and IT component vendors if: (a) future manufacturing cells can be readily constructed from reusable and modular building blocks; (b) means are readily available to establish and maintain flexible (i.e. ‘soft’ links) between the reusable building blocks used in a cell; and (c) the interoperation of the building blocks in a cell can be aligned to business and production requirements in a rapid and effective way, even where those requirements change unpredictably.

It was assumed therefore that future approaches to the design and construction of cell control systems should be able to provide such benefits for both vendors and end-users by bridging the gap between a top-down modelling approach and a bottom-up operation formalisation. This might be in part achieved by separating cell behaviour (i.e. actual dynamic operation of the cell) from control knowledge and system functionality, and also by explicitly representing flexible mappings between conceptual requirements and alternative system behaviours. However, such an approach would need to assume the availability of a suitable set of reusable control modules that can be adapted to meet various end user purposes, i.e. have an inherent change capability.

The paragraph above encapsulates the premise on which this PhD study is based. Hence, the research and testing of ideas in this thesis seeks to prove and/or disprove that premise within the confines of an academic project. However, the study was also assisted via an input of knowledge and data from industrial companies collaborating with the MSI Research Institute at Loughborough University.

The realisation of need for the new approach to the life-cycle engineering of manufacturing cells has structured the primary vision of this research on the overall solutions for the design and construction of manufacturing cell systems which will be developed, implemented, tested and assessed within this research framework.
The approach proposed by this research is characterised below and illustrated by figure 3.2:

a) Use of an enterprise engineering framework to provide an overview structure to formalisation of the requirements of manufacturing cells in terms of the activities, resources and control elements, as well as means of representing the requirements in form of the modelling methods, techniques and tools implemented during the design, construction and operation of the cells.

b) Use of a component-based approach to conceptualise design methods and explicit definitions of cell architecture, control elements and resources used to construct cells in a modular (and therefore potentially reusable) manner.

c) Use of a process-oriented modelling approach to capture and formalise the flow of activities needed within cells, and therefore having means of modifying operational behaviour.

d) Use of a supportive information model to structure use of information during the life cycle in manufacturing cells.

e) Use of a set of integration services to provide connectivity and communications among the elements of the cell systems, in a change capable and scalable way.

f) Use of a model-driven software component architecture to explicitly structure implementation of cells and operationalize interaction and communication between functional modules (e.g. reusable building blocks or components) in a consistent way and align with cell activities and tasks defined via (c).
g) Use of a prototype semi-generic model of manufacturing cells describing common activities and activity relationships in cells, software components (that can be selected and configured to produce a specific “control system” a particular cell), support tools and resource elements (that are assigned to a particular cell) and their interrelationships in a given application domain. The main purpose of this semi-generic model is to provide a modelling structure that establishes a base level of design expertise during the design of component-based cells. This model reconfigures the modelling elements and their relationships and rapidly develops a new specifically defined model for particular domain of cells. Here it was assumed that a particular cell must be a “constituent member” of the broader application domain covered by the semi-generic model of cells [17].

h) a set of modelling and model enactment tools that support the selection, implementation, interoperation and ongoing change of software components used to organise, co-ordinate, control and monitor the operation of a cell.

3.3) Research Objectives

Bearing in mind findings of the literature review and the requirements discussed in the foregoing, the overall objectives of this research study has been to investigate means of advancing state-of-the-art in the provision of methods and tools to support (1) the design of manufacturing cells, and (2) the reconfiguration of manufacturing cell control (MCC) systems from reusable control system components. It was understood that the new approach developed should be appropriate for use with respect to different types of manufacturing cell and yet be able to handle conceptual design, detailed design and real system execution issues of a complex nature. The new approach should lead to a selection and organised use of resources, the realisation of cell processes as required and an analysis of results.

Primarily this work is aimed to:

1) Propose a semi-generic and re-configurable modelling structure to explicitly represent manufacturing cells at a conceptual level of modelling.

2) Develop and test the use of a methodology that provides a modular structure suitable for use at the design level of modelling of manufacturing cells and manufacturing cell control systems.

3) Develop methods and techniques to define specifications and be able to generate an appropriate set of software components to operationalize the modular functionality specified by the semi-generic model.

This work also seeks to achieve a set of secondary objectives including:

4) Develop and implement a generically defined information modelling structure (and supporting information systems) that support cell control systems elements both at the modelling and the operational environment.

5) Specify and implement a set of prototype infrastructure services (e.g. integration, connectivity, and accessibility systems) required during the life cycle of manufacturing cells.
6) Produce a proof of concept set of components in the form of application tools that demonstrate and examine the practicability of the approach to develop change capable manufacturing cells.

3.4) Research Domains

This study was expected to advance aspects of the design and construction of Computer Integrated Manufacturing systems within a specific application domain as illustrated by figure 3.3. The work was restricted largely to a study of internal cell control issues. It only considered in broad terms inter-relationships with other manufacturing systems in a host environment (such as those systems illustrated by figure 3.1).

This work sought to build upon and extend several existing approaches to enterprise engineering, but was focused on developing solutions pertaining to cell control problems. This decision was made with the aim of improving the utility of the models of cells developed during the study. A particular aim has been to develop models from function/behaviour and information viewpoints in a manner that is consistent with models of other systems deployed in a host environment (resource and organisation views of cells have also been developed, but to a lesser extent). Each view needs to be analysed at a certain level of generality with respect to specific phases of the life cycle of manufacturing cell control system, as depicted in figure 3.4.

In order to achieve useful research results in a limited time, it was not proposed to seek to develop and test a very general model of cells to the extent that it can be used in different manufacturing domains. Rather it was deemed to be necessary to constrain modelling to a limited application domain. The application domain chosen corresponded to cellular manufacture of PCB products. However, it was intended that the approach to such semi-generic modelling with further development effort could be extended to generate so-called generic models of manufacturing cells.
With respect to the provision of modelling support for various life phases of cell systems, it was decided that little benefit would be gained from conceiving and developing partial models of cells for use during implementation and execution life phases of cells because of difficulties in pre-defining the requirements and elements of models at this level. Therefore, it was proposed that any semi-generic modelling support for such issues would be for the detailed design life phase, as discussed in chapter 4.

3.5) Anticipated Benefits
As a consequence of realising the approach to cell design and reconfiguration outlined in section 3.2 an aim of the study was to provide proof-of-concept illustration of the following benefits:

- an advance in the modelling of manufacturing cell systems leading to a practical implementation of modelling tools.
- improved overall control over system resources and system operations, leading to more efficient and more appropriate resource management.
- a decrease in time and cost when realising a system design as a result of deploying a model-driven component-based (reusable and modular) approach.
- reduced time and effort needed to reconfigure a cell system in the event of changing requirements and cell conditions.
- ability to extend and integrate manufacturing cells and control systems in a stepwise, flexible, and extendable way.

Summary
Some of the problems regarding manufacturing cells have been discussed along with an overview of possible solutions to those problems. The objectives of this research have also been discussed and restricted domains of work were specified.

The major conceptual concern of this research is to find ways of bridging the gap between conceptual design and system operation, and to improve the utility and configurability of previously available modelling approaches in the target application domain.

The overall research approach was explained as developing a model-driven component-based concept and methodology to design and construct manufacturing cell systems.
Chapter 4

Design and Reconfiguration of Manufacturing Cells - The Research Scope

4.0) Introduction

This research study seeks to develop a new approach to a) the conceptual design of manufacturing cells and b) the detailed design and reconfiguration of manufacturing cell control (MCC) systems. The new approach aims to unify the use of bottom-up techniques, typically used to engineer a specific industrial manufacturing cell, and top-down conceptualisation methods, used formally to define generic descriptions of manufacturing cells.

An overview of related research objectives and concepts were discussed in chapter 3. In this chapter the research approach will be defined in greater detail and the methodologies used to achieve the research goals will be discussed.

To harmonise academic and industrial goals, the research was based on an understanding of common design and reconfiguration requirements, elicited when visiting production and test cells of the D2D (Design to Distribution) Company. D2D is a fairly large manufacturer of Printed Circuit Board (PCB) and the company is located in the UK.

4.0.1) Bottom-Up Approaches

D2D produces, assembles, and tests printed circuit boards (PCB), primarily for the computer industry. Generic cell control requirements were observed during several visits and discussions held with company personnel. The production area included three major sections (that are referred as cells), namely: manufacturing, assembly, and test cells. The cells operate largely asynchronously and autonomously, but workflow between cells is planned and co-ordinated via the use of a MRP system. Many of the cell activities are performed manually, although some automated machinery is used to achieve main stream processes, such as parts-kitting, insertion and onsertion of PCB components and various forms of PCB testing. Information flows very largely are paper-based and requires preparation activities the night before production operations are performed. A central and remote MRP/Scheduling system generates work plans for section managers. The work plans are distributed to cell supervisors for subsequent dispatching of jobs and tasks to elements of the working cells. Therefore, supervisors are responsible for job and task distribution and allocation within the cells. No supporting computer tools are used within the cells. The process of work distribution from the MRP/Scheduler system to the cell operators is known to be slow and its operation requires experienced individuals. In this domain of the company the lack of a consistent and up to date information system to support intra-cell decision-making was clearly observed.
It was stated that the company requirements are constantly changing, and most predictable sources of operational changes are driven by specific requirements of the new orders. The reasons given for not using local schedulers within cells or supportive computerised integration systems was that available systems were found to much increase the complexity of cells and their operations and seldom would lead to flexible operation in event of changes. Whereas changes often occur very frequently with an unpredictable nature.

Initially, this situation was a primary motivation for research that sought to develop supportive computer application tools capable of improving the distribution and allocation of jobs and tasks in a cell based on a general knowledge of resource availability, priorities, integration requirements with respect to the other systems in the company. Therefore, the initial study was focused on developing a change-capable cell control system to expand the degree of practical automation used to control and realise operations within the "Test Cell", but the aim was also to be able to develop and deploy the same methods and techniques of other cells in the company. A set of computer application tools were designed and developed in bottom-up systems engineering manner. This led to the specification and development of a lab-based prototype evaluation system. The detail of this experimental prototype is reported in Appendix D of this thesis.

However, it was soon realised that this initial solution focussed on specific problems of company cells and therefore could not lead to the level of flexibility required to technically support re-engineering of cells in a dynamic and fast-changing manufacturing environment. Rather, it was concluded that the detailed investigation of a narrow scope problem would not be able to establish a consistent interaction method with the other manufacturing systems and cells in a company. Therefore, in parallel a study of more generic solutions to similar problems across different industry sectors was considered.

4.0.2) Top-Down Approaches

As reported above, a bottom-up system engineering approach was used to develop the initial lab-based prototype evaluation system. The application software generated in this way was shown to improve the operation of the particular test cell in certain respects. However, it became evident that the choice of structural relationships between cell system entities and the choice of their method of interaction (both within the cell and with other systems in the manufacturing enterprise) could impact significantly on the extent to which and the flexibility with which supportive software tools can be introduced and deployed within a cell. This emphasised the importance of being able to readily accommodate changes to the cell in the event of influential changes occurring within a host enterprise. It was therefore concluded that it would be necessary to formally define cell requirements within the context of a broader definition of enterprise requirements. It follows that it

---

1 When this study was initiated, the D2D manufacturing Company was a part of the ICL enterprise. More recently Celesitica acquired the D2D Company.
was considered necessary to specify cell requirements and conceptual designs via the use of a top-down modelling approach that complements the bottom-up system implementation methods adopted. It was assumed that the modelling approach chosen should facilitate the holistic design of a cell in a manner that is consistent with the detailed design and configuration of cell control systems from reusable control systems components. By formally modelling cell requirements and configurations of control system components it was also assumed that changes in requirements could readily be mapped onto new configurations of control system components.

4.1) The Research Concept
The overall research concept was discussed earlier in chapter 3 and illustrated by figure 3.2. The research concept is based on capturing common requirements (e.g. resources, activities) of a particular domain of manufacturing cells in a semi-generic manner. The requirements should be formalised using appropriate modelling methods and tools in the form of generic functional components (and their associated resource and information objects). Finally, the functional components should prescribe the development and generation of a set of software components capable of constructing an actual set of cell control application tools.

In this scenario three main conceptual requirements have been considered as important characteristics which the research solution should support. The conceptual requirements are namely:

- **Reconfigurability** – The ability to flexibly and consistently change the configuration of the elements within the cell systems corresponding to future changes that will occur in the cell specifications and goals.

- **Reusability** – The manufacturing cells (and the cell control systems) should be designed in a generic and modular manner to inherently facilitate re-use of system components.

- **Generality** – The design approach should be developed for a range of similar manufacturing cell systems. This should enable the developed systems to be versatile and reusable in a variety of manufacturing cells and consequently reduce the design time and cost.

Figure 4.1 illustrates an overview of the research concept developed in this study. The figure includes several sections, each being associated with a main target of the research (as described in §3.3). The way that each section achieves its target will be discussed in greater detail in chapters 5 to 7.

A brief description follows about each section of figure 4.1.

- **Semi-Generic Model of Manufacturing Cells**

  The main element of this model was expected to be a set of reference tasks that are common in the domain of study. This model should formally describe the reference tasks. There should be a capability to add or remove tasks so that the model can represent different classes of manufacturing cell. It was assumed that the semi-generic model should also include reference
descriptions of the resources required to carry out the reference tasks. The resource descriptions need also to be described in a formal and reusable manner so that the design of a specific cell can begin to assign particular resources to particular tasks in an organised and sensible manner. It was also assumed that the tasks, resources, and associated methods should be organised in a reference library. This might be in the form of implementation modules capable of realising various particular models that correspond to a specific sequence of cell operations (that encode a given set of production requirements from the cell).

- **Configured Model of a Particular Cell**

As a result of re-configuring and instantiating the semi-generic model it was assumed that a particular model can be constructed that inherits properties defined when specifying generic operating requirements and suitable resource sets. However, it was assumed that it will also be necessary to retrieve information on real cell conditions (e.g. product engineering data, new orders) to construct a particular cell model in detail (that formally describes the production of a particular product or group of products (of specific quantity, size, due-date, etc.). In addition it was assumed that it would be necessary to establish links between particular tasks and methods and the resources they require. Finally it was assumed that a particular model should be developed to support the configuration of technical applications (that would most probably be realised by software processes and computer hardware) to integrate the operation of system entities by utilising the services of an infrastructure system.

To facilitate change (of various kinds) it was also decided that the particular model generated should be readily modifiable, although it was assumed that reconfiguration of the particular model should be achieved with due reference made to the semi-generic model.

- **Component-based Generation of Software Modules**

A set of reusable software modules (or components) should be specified and generated to provide run-time execution capability for the "functional components" defined by the semi-generic model. It was decided that some form of component-based approach to the design, construction and implementation of software modules should be deployed. The cell control applications (so-called in this study "Cell Application Tools" - CATools) are assembled from a set of software modules (along with appropriate user interface and interaction functions).

- **Supportive Information System**

It was also assumed that a semi-generic information model would need to be developed with a capability to provide the information required during each life-phase of a cell system, from requirements planning through to the real world execution phase.
The supporting information model should define a set of information entities needed to underpin the specification, assignment and operation of tasks (categorised into classes). It was found that information entities contained in the supporting information model should be purely data-oriented and maintained completely separately from functional modules. This was assumed to be necessary to improve the configurability of the models produced.

It was also found necessary to develop data manipulation methods and tools (i.e. essentially an information system for a particular cell) to realise physical connections between resource entities and the data storage system. These methods and tools should be designed in a form that enables them to handle data requirements during the run-time execution life phase.
• **Prove-of-Concept Application Tools**

It was assumed to be necessary to develop a number of application tools (i.e. software tools) with capabilities to implement and test the theoretical concepts proposed by this work. It was understood that these tools might be deployed from existing tools (e.g. commercial tools or tools developed by other research groups) or might need to be developed in this research. The adopted tools can assist formalisation of the system requirements (e.g. SEWOSA Case Tool) and also modelling of information system (e.g. EXPRESS Tools). The developed tools may be organised into different sets. A first set of these tools would need to provide user interface facilities for building and developing the semi-generic model to enable its use by system developers (so-called in this study "Cell Design Tool - CDTool). A second set of these tools should be designed and built for use by system end-users in support of system execution (including CATools). It was also intended that the second set of tools would be automatically generated and configured by the first set of tools to facilitate reconfiguration of the cell. It was assumed that industry would want an option to use certain pre-existing, well proven tools with which they are familiar and have invested in. Hence a need to provide a capability to connect to other software systems (including data storage systems) in a host environment.

4.2) ** Adopted Methodologies**

The overall research methodology was conceptually discussed earlier in this chapter. In addition to the methodologies developed in this research, the research was intended to deploy (and improve as required) some of the existing well-established and standard approaches to the design and implementation of manufacturing enterprises. These include:

- **Enterprise modelling architecture and methodology** – A modelling architecture should be deployed as backbone of the model-driven component based approach studied in this research. A study of available modelling architectures was made as explained in the literature review (§2.3). This research has its foundation on the CIMOSA modelling architecture [56] that has been extended and operationalized [17, 136] to structure the design and implementation of enterprise systems. Choosing CIMOSA was mainly based on its proven track record with respect to the conceptual design of systems (including manufacturing cells) and its deployment by a large number of manufacturing end user companies [137]. Despite a number of shortcomings (see §2.7), the expandability of this architecture provides a comprehensive modelling capabilities in support of a top-down approach to the design and analysing of manufacturing systems. Ongoing international effort on developing **Generic Enterprise Reference Architectures and Methodology (GERAM)** [62], based largely upon the concepts and framework embedded into the CIMOSA architecture. Therefore, adopting the CIMOSA architecture allowed the research study to structure and expand the system design based on the internationally agreed concepts.
• **Enterprise Integration Systems** – Enterprise integration systems are largely used to facilitate the transfer and sharing of information throughout an enterprise. It should provide the right information, at the right time, in the right place [54]. Design and implementation of the EI raises issues such as change of information model, system networking, information exchange format and the used a suitable enterprise modelling approach. Much effort world-wide has been directed toward developing methods, framework and technologies which support the realisation of fully integrated systems. The outcome has been a wide variety of supporting tools in the form of theories, methodologies, application tools, techniques, and modelling tools (see literature review §2.5.1). In this research the CIMBIOSYS [138, 139] tool (also see §2.5.2) was used to establish interaction among the system entities and to transfer information. Bearing in mind that development of a comprehensive enterprise integration system is not the major concern of this research, the main reasons of choosing CIMBIOSYS were: a) previous experiences of MSI research institute members on developing and implementing CIMBIOSYS, b) ease of use and compatibility with the other tools used in this study. The main roles of CIMBIOSYS services in this research are to facilitate and organise data transfer amongst the elements of cell control applications, and provide definitions of embedded functions within the software components to enable these components to interact with the infrastructure services (e.g. network and information system).

• **Information modelling** – Information modelling is concerned with the construction of symbolic structures that capture the meaning of information and organise it in understandable ways [67]. To provide consistent information and deliver it to support the modelling and real-world life cycles, and to share knowledge within a complex system, a formal information model and its corresponding information system were required in this research. Initially a number of information modelling approaches were reviewed (see literature review §2.5.1). The developed information model had to be compatible with the existing manufacturing communication system, interaction and data exchanging methods selected, and also support working with legacy systems. Therefore, information modelling method was aligned with standard modelling methodologies such as STEP Standards [101] and information modelling languages such as EXPRESS [99] where they were applicable. Background research and tools developed by researchers in MSI Research Institute [140, 141] regarding the information modelling were considered in this respect².

• **Component-based approaches** - Potentially, component-based (CB) approaches can facilitate the detailed design and configuration of change capable manufacturing systems. The basic

---

² Here, it was envisaged that application of the standards modelling methods would be most suitable at a detailed level of modelling, and in some extent, is not able to sufficiently support a generic level of the modelling. Therefore, the standard methods (e.g. STEP/EXPRESS information modelling method) have not been recommended (and used) in the formalising of the conceptual design of the modelling system (see §5.3.2).
concept of the CB approach is that it relies on the reusability of generically defined “system functional modules” (hardware or software modules) during the design and operation processes. The functional modules should be represented and operationalized in form of software components during the implementation of the system. Component-based approaches to software system engineering have emerged based on the use of object-distributed technology and the software engineering methods (see §2.6 in the literature review) primarily with the aim of improving the reusability of the software components.

A component-based approach has been developed and employed in this research, leading to the definition, reuse, and re-configuration of cell "functional components" (This is further discussed in §6.5 - RACM approach). Emerging software engineering principles have been adopted during the design and generation of the software components and the graphical user interfaces required to develop prototype application tools and to test and validate their use.

4.3) The Research Plan
The need to develop improved means of engineering manufacturing systems was taken as the basis of this research. The research area was restricted to problems involved in manufacturing cell control domains. It was discussed that the current status of engineering manufacturing cell systems (referred as the AS-IS situation) can be improved by developing a reconfigurable design and change-capable construction approach (referred as the TO-BE situation).

The research background, literature review, and state-of-the-art approaches were reviewed in chapter two. Also a more detailed discussion of problems, research objectives, and potential solution is included in chapter three. Whereas this chapter specifies the research scope more exactly by discussing details of the research approach, its technical objectives, research methodologies, and the primary assumptions made.

The research was planned in three stages as illustrated by Figure 4.2.

At the first stage, a semi-generic model of manufacturing cell systems was conceptualised and the methods and techniques required to implement the model were designed and constructed. Here it was intended that research would be based on the use of three complementing approaches namely: enterprise modelling approach; component-based approach; and emerging standard methods to model information systems. Essentially, two different methodologies were developed and used to facilitate the modelling of manufacturing cell systems. These include: a) development and use of the SEWOSA method during the requirement and concept design of cells, b) design, specification and development of the “Resource Allocation and Configuration Method (RACM)” to implement component-based system engineering concepts at the detailed design level.
# The Research Plan

## Manufacturing Cell Systems - AS-IS Condition

### Enterprise Modelling Approach - CIMOSA Architecture

Conceptualising Semi-Generic Model of Mfg. Cell Systems

- Defining Generic Tasks (functions)
- Application of SEWOSA method to represent the functional requirements of cells at the semi-generic level
- Generic information requirements

### Component-based Approach to the Design and Configuration of Cells

Resource Allocation and Configuration Method (RACM)

- Definition of the generic functional entities of the cell
- CIMOSA modelling structure
- Specifying the functional components (incl. info, resource, processes)
- Building library of the reusable building blocks (e.g. Software Components)
- Testing the SEWOSA method at the design level
- Developing sample set of software components

Information Modelling Approach - EXPRESS Modelling Language

Developing Semi-Generic Information Model of Mfg Cell System

- 3-Schema Model - CIMOSA/EXPRESS Mapping
- EXPRESS Model
- STEP Standard Data Transition Technique

### Change-Capable and Reconfigurable Manufacturing Cells - TO-BE Condition

Modelling at Particular level

- Instantiating the semi-generic model for specific cases

Developing the Prototype Applications

- Developing Cell Design Tool (CDTool)
- Developing Cell Application Tools (CATools)

Test Case

- Modelling Mfg Cell for Producing High-Pressure Cylinders

## Analysing the Research Achievements

### Analysing the Results

- Evaluating the concepts
- Analyzing the Case study results
- Assessment of the research approach

### Conclusions

- Specifying the Achievements
- Discussions and Recommendations

## Research Conclusion

Figure 4.2: Overview of the Research Plan

During the second stage of the research, further research was planned to test the viability of concepts selected, developed and conceived. For enterprise modelling and component-based approaches the use of a semi-generic model of cells was tested at the so-called particular level of modelling [56]. The aim was to develop techniques to readily reconfigure the semi-generic model
and generate corresponding particular models that accurately encode real-world cases. Also the
development of a set of prototype application tools was planned which can be used within a
modelling environment to test the concepts. Further evaluation of project methods and tools
developed by the research would be centred on case study to assess the practicability of the models.
During final stage of research, it was planned that results would be analysed and project
achievements assessed in the light of the original objectives and expected benefits.

4.5) Assumptions and Definitions
When defining and developing the new approaches and tools a number of general restrictions were
imposed and assumptions made as follows.

- **Life Cycle Engineering Restrictions** – The research has focused on structuring and
  supporting: a) requirements specification and conceptual design life phases of manufacturing
  cells and, b) the detailed design, implementation description and software configuration (or
generation) life phases of manufacturing cell control systems. Therefore, although the
  conceptual modelling of complete cells is carried out, during the physical realisation of those
  conceptual designs a sharper focus has been placed on building cell control systems from well
  defined and reusable software components.

- **View Restrictions** – This research has considered manufacturing cells from various
  perspectives (or modelling viewpoints) including function, information, organisation, control,
  resource, and economic viewpoints. However the prototype modelling tools developed and
tested during this research study have been focused on the first two of these viewpoints mainly
  for pragmatic reasons, i.e. by supporting these views it was assumed that maximum benefit
  would be gained within constraints imposed by the limited project resource available.

- **Domain Restrictions** – The main application focus has been on manufacturing cells with a
  capability to carry out processing tasks on a number of products in small to medium sized
  batches. Here the products processed by a cell may belong to a family or may be quite
different. Their main shared attribute is that they require a similar set of processing tasks to be
carried out on them.

Further domain restrictions have occurred naturally as the project has focused initially on
manufacturing cells used during the manufacture of printed circuit boards and in later case
study work in machining cells used to produce hydraulic products. Therefore the types of
processing tasks studied the kind of resource sets required and the kind of control system
elements developed have been even more sharply focused by specific application requirements.

Further domain restrictions have occurred with respect to the simplification of models of
processing tasks. For example when producing abstract models of generic tasks (common to
the application domain studied), it was considered to be necessary to ignore the effect of certain
variables (such as materials handling or tooling issues related to a type of task) to realise a modelling scope that does not overly restrict the general applicability, extensibility or reusability of the models developed. However it was understood that by limiting the completeness of a model it might threaten the practicability of its use in the life cycle engineering of a specific cell.

For instance, a model may formalise machining processes in a segment of a cell and to define related monitoring requirements. This model may allow for certain variations in terms of product properties (e.g. geometry, tolerances, and colour of the part). The model may contain a descriptive definition of the process (e.g. process-plan) that describes a fixed process. That is independent of the behaviour of other enterprise entities, and often independent with respect to the timing of events. This static type of information can be used to model a fixed process. However, any change in sequencing of the activities that attribute processes, resources to be used, and the methods to be deployed, will require a new descriptive definition to cover the new circumstances. Despite optional capabilities provided by certain pre-existing enterprise engineering approaches [142], it was understood that formalising cell processes in a more complete and dynamic manner would likely scale up the problems. From this kind of reasoning it was concluded that a more generic level of description was required to support modelling of manufacturing cell processes and that at this level suitable separations of issues would need to be determined to minimise where possible effects of exploding complexity. It was also assumed that at a complementary detailed level of modelling the level of description could not be inappropriately limited, otherwise it would not prove possible to retain an adequate “connectivity” between modelled and real control systems elements used in a specific manufacturing cell.

- **Generic vs. Partial Model** – The term “generic” has been used in different manufacturing contexts with various understandings and definitions. This research uses the word “generic” in conformity with the definition suggested by the CIMOSA consortium [56]. However it was understood that it might not prove practical to develop generic models of cells that cover all important aspects of them in a generalised way. Therefore, where generic models can be developed (or where certain fragments of such models can be developed) it was assumed that their main use would be to serve as guides or templates to initiate design processes, e.g. where conceptual thinking is required with reference to certain abstract concepts. Clearly generic modelling can be expected to focus on a limited subset of issues of concern to would-be users and therefore would hide or lose certain detailed information. Also complete (or holistic) generic modelling (taking account of all possible abstract foci of concern) was assumed to be likely to involve a prohibitively expensive set of modelling activities. Consequently, the utility of generic models was expected to be limited.

In the remainder of this thesis the terms “semi-generic” model or partial model will refer to
domain-specific models that can be reused in the domain concerned without any major engineering effort. It was assumed however that the extent to which a semi-generic model could be reused in different manufacturing domains would be dependent on the type of model concerned, for example a task model might find a more narrow domain of application than a resource model.

- **Generic vs. Reference Model** – From much of the enterprise modelling literature, it is difficult to explicitly determine a difference between the use of term “reference model” and “generic model”. In this study, a reference model will be used to represent a set of reusable elements (possibly in form of a library) including for example tasks, methods, and software components, which might be logically related and interacted. The term might also refer to a commonly accepted (or standard) model. Whereas a generic model will typically specify the ways in which a set of reusable elements may work together. In fact, a reference model may be used to construct a generic (or partial) model. However, “reference models” may also imply different concept within the commercial sectors.

- **Generic Tasks** – In this research it is assumed that cell processes comprise a set of activities that need to be carried out to realise the purpose of a cell. Models of cell processes can be generalised in different ways and from various perspectives. By adopting a functional view, cell processes (activities, operations, and tasks in general) can be categorised with respect to physical resource capabilities, e.g. drilling process, milling process, or soldering process (functional decomposition- see sections 2.1 & G.2). By adopting use of a hierarchical organisational structure, relating activities, tasks, and processes, it may prove practical to define a set of generic (and even standard) tasks. It also assumed that the definition of such a category of generic tasks might not be fully independent of the device capabilities required to carry the tasks out (e.g. a drilling task could be carried out on a drill, mill or a machining centre, but not on a soldering machine). GRAI Laboratory [76] and NIST [49] publications have considered this type of categorisation.

Recent enterprise modelling approaches have categorised cell tasks based on a consideration of the processes carried out within a cell [37, 120]. Previous work indicates that the type of modelling should be linked to granularity of the process models produced [36, 53]. At a particular level of modelling a fixed production process in a cell (such as “producing concrete”) might be modelled (as mentioned by Dobuis [114]) whereas at a generic level of modelling a reusable task can be modelled such as “adding water” and “adding cement” and a “mixing

---

3 In CIMOSA documentation, a generic model has also been termed as “reference catalogue” [56]

4 There is again confusion in literature over the use of terms “Reference Model” in commercial sectors. Some commercial application developers use the particular manufacturing specification standards such as "standards for machinery tools" [143] to develop comprehensive tools to support some degree of industrial changes. It is believed that these products cannot necessarily be called reference model without specifying the particular domain in which they can be valid.
process". At an even more generic level of modelling, generic tasks can be related and organised as elements of a higher level of process abstraction. Such a high-level of process, so called core functions [51] concentrate on the type of the processes like: scheduling jobs; controlling process quality; and monitoring status, as also mentioned by Williams [52], Vernadat [19], and Camarinha [44].

It was concluded that the definition of generic tasks based on some process decomposition requires a consistent approach that covers particular, semi-generic and generic types to enable the process models to be defined, reused, and expanded.

In this research, it was planned that generic tasks would be defined based on a decomposition of common processes carried out during the batch manufacturing of PCB products.

Summary
This chapter considered general findings from an initial bottom-up approach to solving cell control problems in an industrial company. This led to a definition of need for a complementary top-down modelling approach. Therefore the need for a semi-generic modelling methodology was considered in general terms. General characteristic properties of such an approach were assumed to be inherent: reconfigurability, reusability, and generality.

Also described in this chapter was a rationale for deploying appropriate: enterprise modelling and integration methods; modelling tools; concepts developed with respect to component-based technology and software engineering.

The research plan was also discussed. Here it was proposed that the research should be carried out in 3 stages, namely: (1) semi-generic modelling of manufacturing cell control systems; (2) evaluation of the utility of the semi-generic model as a means of structuring and supporting the modelling of particular cells, and development and use of prototype tools that facilitate detailed modelling of cell control elements and realise their configuration and operation within a runtime cell; and (3) analysing the results and conclusions.

Finally the importance of some of the restrictions and simplifications made when modelling manufacturing cells were discussed.
Chapter 5

Developing Semi-Generic Modelling Concept for Manufacturing Cells

5.0) Introduction

This research seeks to establish links between bottom-up and top-down approaches to engineering cells to align the operation of cells to enterprise goals and requirements whilst realising an improved change capability.

The research approach was described in section 4.1 and centred on the development of semi-generic modelling capabilities that support the component-based design and construction of manufacturing cells. To define semi-generic requirement of cell systems, the research sought to characterise common system requirements in terms of common tasks carried out in a specific domain of manufacturing cell systems, namely the PCB manufacturing domain. These common task requirements must be formally represented and applied within a suitable modelling framework. It was intended that abstract and explicit definitions of common requirements developed and used in this way could facilitate the definition of specific requirements for a particular cell.

This chapter develops a model of the conceptual requirements of manufacturing cells and manufacturing control (MCC) systems at a semi-generic level, corresponding to the CIMOSA partial modelling level. Here function and information aspects of cells and cell controllers are the centre of attention.

5.1) Semi-Generic Modelling Concepts

Following an assessment of potential uses of cell control applications tools that were developed for use as part of a bottom-up approach to engineering a specific industrial cell system (see § 4.0.1), it was concluded that an improved design and construction approach should explicitly formalise requirements, system design and components in a change-capable manufacturing cell system (see § 4.0.2). Therefore as part of the research it was deemed to be necessary to define and capture one or more semi-generic models of manufacturing cells where these models provide an exemplar modelling structure. This structure was to be used to target the conceptual and detailed modelling of manufacturing cells in a given application domain. A key requirement here was the need to be able to reuse the model to guide system designers and constructors about many different instances of cells, each designed to meet a particular set of manufacturing requirements and to utilise a particular set of control system components and resources. Thus a semi-generic modelling structure was defined and used as explained below. The specification of this semi-generic model was based on a study of common requirements and solutions found within a number of printed circuit boards (PCB) manufacturing cells deployed by the D2D Company.
First a set of common tasks carried out in PCB manufacturing cells were identified and represented using computer executable modelling constructs. Here a decision was made to formally define models of tasks as a set of "generic activities" and their interrelationships, corresponding to the CIMOSA modelling approach deployed in this research. This decision was predicated on the fact that proven CIMOSA representational concepts were already available to model relationships between activities.

As illustrated by figure 5.1, the semi-generic model of manufacturing cells was defined in two parts, namely the "Conceptual Design level" and "Detailed Design level" of modelling. The first part corresponds to the requirements definition level of the CIMOSA modelling structure. It models generic activities (i.e. common tasks) in the form of Business Entities (BE), including generic models of Domain Processes (DP), Business Processes (BP) and Enterprise Activities (EA). This part of the semi-generic model conceptualises aspects of common requirements to guide the design of manufacturing cells, in terms of their function, information, resource and organisation aspects. However as mentioned before, in this study the major focus of attention was centred on developing function and information views. In addition, the first part of the model also includes "Reference Procedural Rules". These rules can be used to define sets of common logical relationships between activity flows during the operation (runtime) phase of manufacturing cells. The conceptual design part of the semi-generic model also includes definitions of generic information requirements related to defined generic activities.

The second part of the semi-generic modelling structure (i.e. the Detailed Design modelling level) was specified and developed to conform to the design specification and implementation description.
levels of the CIMOSA modelling structure. The aim here was to target and facilitate detailed design in a manner that results in the development of model-driven, component-based solutions. At this stage of modelling, control system entities should be selected and configured to organise, coordinate, control and monitor the interoperation of resources in a cell so that they achieve specific cell requirements. Here specific cell requirements are defined by fleshing out particular instances of generic tasks (as defined during the development of the first part of the modelling structure).

When developing the second part of the semi-generic modelling structure four types of modelling construct were identified as being required to facilitate the modelling and assignment of elemental building blocks of functionality (to be deployed in the runtime cell) to each atomic activity element (i.e. each enterprise activity). These constructs are used during detailed design modelling and include: Function Entity constructs, used to represent the functional capabilities of physical resources in a cell; supportive computer tools; information modelling constructs that are specified during the development of information model created during the detailed design level based on the information requirements defined at the conceptual design modelling level; and software components needed to actually organise, control and monitor the operation of the generic tasks by resource elements assigned to the cell.

Although it was understood that there would be potential benefits if the modelling concepts developed during this study remained as far as possible conformant with the CIMOSA modelling concepts, some of the modelling approaches proposed by CIMOSA were not found to be appropriate (or sufficiently well defined) for use to describe the application of semi-generic model at the detailed design level. Therefore, the following modifications were applied to the CIMOSA modelling approach deployed in this research.

The design specifications that lead to execution of the model, later at the particular level of modelling, were proposed in this study as part of the detailed design level of the semi-generic model. It was however assumed that a separated implementation level of semi-generic model (similar to the CIMOSA structure) may not be able to provide sufficient detailed descriptions to directly implement the methods, defined at the design level.

In addition, when using an explicit definition of a requirement specification to guide the conceptual and detailed design of a particular cell in conformance with CIMOSA concepts it was understood that it would be necessary to assign functional components to enterprise activities including: resources (with a capability to realise one or more activities that constitute elemental tasks); control processes (which ensure that the collective runtime operation of resources fulfils specified requirements); and supporting processes and components (that underpin necessary interactions between control processes and cell resources).

However, currently CIMOSA concepts do not themselves target system development activities towards the realisation of component-based cell controllers. Therefore a key design assumption
made was that it would prove to be possible to develop and use new concepts (that complement existing CIMOSA concepts) where the new concepts target system design and construction towards the selection and use of set of formally defined system components at the detailed design level. The detailed design part of the semi-generic model and the implementation issues is left to be discussed in chapter 6.

5.2) Generic Task and Cell Groupings

As explained in section 5.1, a basic assumption made during this study was that modelling should be centred on a definition of value added processes in a cell that is expressed as a set of "tasks" to be carried out within a given manufacturing cell domain. Here it was assumed that a "task" can be carried out either by a machine or by a human (or combination of the two). In different domains for example a generic task could include: turning a part, setting-up a machine, transporting material within or out of cell, testing of a printed circuit board, or carrying out quality checks and measurements. Therefore, tasks can be done manually or semi-automatically. In this context a task can be viewed as being a set of physical operations carried out to achieve a specific purpose that normally will add value by changing the state of products in a cell.

As discussed and concluded earlier (the literature review §2.1.1), one categorisation of generic tasks in typical manufacturing cell systems comprises: 1) receiving orders, analysing and distributing tasks, and allocating resources; 2) operation of tasks; and 3) monitoring tasks.

It was understood that in different cell domains the generic tasks might be modified. For instance, in a particular PCB manufacturing test cell, tasks may typically be divided into different functional groups such as conducting a test process, controlling material flow in a cell, providing instructions to operators, providing tools, and so on. Each task might generate a need for other tasks such as the spawning of a material transportation task from a main machinery task.

In many practical cells a task or group of tasks may need to be carried out by an appropriate section of the cell. For example, to carry out the printing of a large number of Printed Circuit Boards (PCB's) one section of a cell may be assigned to the role of printing, one may be used to convey material (or semi-finished jobs) and another for washing parts. In this study these various sections of a cell were considered to be a "groups".

A "group" in a cell includes a set of tasks, required resources, infrastructure services and relationships defined and allocated to carry out those tasks.

A cell may comprise just one group or many groups as illustrated by figure 5.2, dependent on requirements and characteristic and properties of cell.

Each "group" will need to be organised so that it can perform specific classes of task in a cell.
Unlike the concept known as Group Technology [135, 144], the notion of "grouping" in this study is based on an observed need to flexibly classify and associate tasks (and resources capable of performing those task) to align with certain cell requirements. In general, the capability to support grouping in cells is aimed at: a) improving the manageability of cell operations, in terms of physical activities and computational control processes, b) increasing the level of autonomy in a cell by distributing activities and responsibilities.

A group could be a series of machines along a production line, which are controlled by a group co-ordinator. The group co-ordinator might be fully automated, (for example a control system for a limited area like the control system of an AGV) or manually controlled by a human operator. In complex groups, a group supervisor might instruct several machines or operators. Generally each group will be responsible for performing a set of tasks by receiving them from a cell supervisor, retrieving the supporting data, and reporting the results of operations. Each group co-ordinator should provide sub-tasks for each element of the group by accessing more global data as required, and also should deal with unexpected incidents that occur within the domain of a group, by making appropriate decisions at the group level.

Consider the case where, group one is doing a “washing up” operation on PCB’s and group two transfers parts to a drier. In the case of a breakdown at the washing up machine (as part of group one), the system cannot finish the job. Therefore, the monitoring system should report the new situation to the cell co-ordinator so that re-scheduling of the system can be carried out by considering alternative resources and possible buffering of material (or parts). Then group two should update its task list with new tasks regarding the breakdown that has occurred in the system.

Figure 5.2: Task grouping concept in the manufacturing cells

---

1 Term ‘Group’ has been used by other researchers to address a set of specially selected physical machines to produce a range of products. For example Burbidge states: “Group is a list of machines required to produce a particular production family of parts”[145]. He also refers to ‘Group’ as organisational units, as he states: “Group Technology is a method of organisation for manufacturing, in which the organisational units (groups) complete all the parts they make at their particular processing stage” [146]. However, as discussed earlier, in this study term ‘Group’ does not only refer to a selection of physical machines. The ‘grouping’ idea, to some extent, has many similarities to the concept of “Holonic” manufacturing introduced and used by some research groups [127, 147, 148]. However, holonic concepts are applied at different level of modelling and control of business processes.
It is assumed that the cell supervisor allocates tasks to individual groups. Subsequently, groups need to retrieve sufficient data to carry out the tasks assigned to them, i.e. by accessing a common data storage system. Within the groups, the tasks may be decomposed into a set of jobs (and required sub-processes of retrieving data for those jobs).

Each individual group should be designed to possess a capability to access supporting tools, ideally via an integrating infrastructure system. Each group should be able to send requests to supporting tools (e.g.: a scheduler or information system that might in practice be external to the cell) and to handle potential conflicts with others groups.

The key property in the “grouping” concept is that responsibilities are split up between different authority sections of a cell by decentralising control processes, so that each section is responsible for coping with new situations or changes in its local domain. This should improve configurability and scalability of the system. In addition, the cell grouping facilitates the development of cell control software applications (i.e. CATools – this will be discussed in chapter 7).

The concept of “grouping” is actually a virtual categorisation of resources and control elements in a cell. Potentially, it should be readily applied in different cell domains and facilitate accommodating changes regarding the resource and control systems. Furthermore, the concept should help cell control systems to be defined in an extendible and modular manner. At an appropriate level of granularity, cell and cell controllers can comprise reusable modules (and their associated software components) so that the system can more readily cope with phased automation, as full automation may not always be necessary or successfully applied.

5.3) Conceptual Design of Semi-Generic Model

As illustrated by figure 4.2 (the research plan described in chapter 4), the CIMOSA modelling architecture was chosen as the first approach of the research framework. This modelling approach provides underlying modelling support suitable for the conceptual design part of the semi-generic model to specify generic functional requirements based on the defined generic tasks in a cell. To capture, formalise and graphically represent the functional viewpoints of the semi-generic model at the conceptual design level, the SEWOSA CASE Tool developed in the MSI Research Institute, Loughborough University [117], was used.

In addition to defining functional requirements, the generic information entities required for the cell should also be specified. These generic information entities would need to be defined and formally structured at the conceptual design level of the semi-generic model in conformance with the CIMOSA information view. It was understood that the information entities should be formalised in the form of computer processable model at the detailed design modelling level by using a standard information modelling method. At this modelling level also the other generic elements of the semi-generic model, generally termed as “system components” (e.g. resource, control process, software
tools) should be specified.

Table 5.1 summarises the approaches, methods and techniques used in this research to develop the semi-generic model. The remainder of this chapter and the next chapter will report on the design and development of a semi-generic model of PCB manufacturing cells at the conceptual design and detailed design levels of modelling.

<table>
<thead>
<tr>
<th>Generic Elements</th>
<th>Approach</th>
<th>Method</th>
<th>Techniques/Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic Tasks</td>
<td>Experimental</td>
<td>Study, experiences, expertise knowledge</td>
<td></td>
</tr>
<tr>
<td>Generic Functional</td>
<td>Enterprise</td>
<td>CIMOSA Architecture</td>
<td>SEWOSA CASE Tool</td>
</tr>
<tr>
<td>Requirements</td>
<td>Modelling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generic Information</td>
<td>Information</td>
<td>CIMOSA &amp; STEP Standards</td>
<td>EXPRESS Language</td>
</tr>
<tr>
<td>Entities</td>
<td>Modelling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generic System</td>
<td>Component-Based Design</td>
<td>RACM method developed in this research and discussed in the next chapter</td>
<td></td>
</tr>
<tr>
<td>Components</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Methods and tools used to develop the semi-generic model of MCC systems

5.3.1) Modelling Functional Requirements

In conformance with the CIMOSA requirements definition modelling level, the conceptual requirements of the semi-generic model were specified at the partial level. According to the CIMOSA concept of instantiation, the requirement should be particularised for specific modelling cases at the particular level of modelling.

At the “conceptual design” level of modelling, it is necessary to concentrate on developing a partial model. According to CIMOSA a model needs to be developed from four different viewpoints viz.: information; function; resource and organisation viewpoints. However because of time constraints on this study, the research project was concentrated on investigating the development and application of function and information views. Figure 5.3 illustrates the modelling steps required at the requirement definition level of modelling.

Therefore, at the first stage of modelling, a
"CIMOSA domain" was established and named the "manufacturing cell system" domain. Figure 5.4 illustrates the semi-generic model of the manufacturing cell system domain, specified during this study and its relationships with other manufacturing system domains. This domain will be one part of a larger manufacturing enterprise domain comprising other CIMOSA-conformant and non-CIMOSA-conformant domains. Therefore in general it is necessary to define and maintain relationships with these other domains.

In this research the details of non-CIMOSA-conformant domains are only partially considered, to the extent of modelling events and object views exchanged with CIMOSA-conformant domains.

A manufacturing cell will typically interact with non-CIMOSA-conformant domains as follows: an engineering department will define activity procedures; a production control system will issue definitions of cell requirements and production orders; a stock control system will manage the consumption of materials; and a maintenance department will provide technical support [56]. Other domains can be attached to the model by defining relationships (i.e. shared events and objects) that exist between them and the modelled domain.

The manufacturing cell domain needs to receive input information about orders and tasks that should be carried out. Also it should generate information about results of cell activities and cell

---

2 The reader should be fully aware that the study focus was not on developing a complete and very widely applicable semi-generic model. Rather the focus was on developing and testing new methodology and concepts aimed at supporting the use of such a level of modelling.

3 This list is not intended to be definitive or complete.
situations. In CIMOSA terms, these information types are termed an input information view and an output information view, respectively.

To completely define a manufacturing cell domain the objectives and constraints pertaining to the domain need to be defined. These include relationships with other non-CIMOSA domains, events, domain processes and declarative rules, enterprise objects, and object views. These specifications were captured in form of *domain templates* and used as a standard building block⁴ of models. A domain template must be populated with all the information needed to fully describe a domain also each element in the domain may be fully described by use of an individual building block in the form of the CIMOSA template (see Appendix B).

Decomposing the modelling domain led to the formal description of the generic tasks in the form of *Domain Processes* (as defined by CIMOSA and implemented by SEWOSA). A *domain process* defines the functionality and behaviour of a given domain. This was generated using the ‘context diagram’ capability of the SEWOSA tool kit. Characteristics of the semi-generic *domain processes* established in this research were as follows:

- **New Task Management** – This *domain process* receives new tasks from a non-CIMOSA-conformant domain and is triggered by external events. It also assesses the capability of the cell to prepare the scheduled lists of tasks, materials, tools and auxiliary equipment. Finally it dispatches tasks to resources having generated a work-plan.

- **Operations Management** - This *domain process* specifies tasks related to cell groups. It prepares data and resources and identifies jobs for each task assigned to groups. Finally it carries out jobs and generates messages to determine the status of tasks and resources.

- **Monitor Cell** – This *domain process* displays conditions of each enterprise entity in a cell domain. The display can be at a requested time or continuously output during run-time execution of a cell. It also analyses results and provides reports by directly accessing static and dynamic data related to the cell domain.

It should be noted that the importance of each *domain process* may vary from domain to domain and may be highly dependent on the certain characteristics of a target cell. For instance, in a milling cell, which produces large cast-iron beds for grinding machines, issues related to the monitoring and receiving orders may not be very critical, whereas the detailed processing of jobs might be of major importance. On the other hand, in a mass production cell designed to make large volumes of simple electronic parts, the job itself may be easy to do but managing the amount of material and maintaining the adequate performance level are likely to be issues of prime importance.

---

⁴ The words *Building Block* has been used in complying with CIMOSA terminology and should not be confused with the terms of “building blocks” or “reusable components” defined earlier.
The next modelling step was to decompose the domain processes into a set of hierarchically structured functions. It was necessary to analyse semi-generic system behaviour in greater detail via further and sequential decomposition of each domain process. This led to an identification of elementary units of activities and their relationships that were expressed as a set of Business Processes and Enterprise Activities. In the manufacturing cell model produced in this way, each domain process was decomposed into business processes and enterprise activities.

The Task Management domain process was decomposed into the following business processes as illustrated by figure 5.5:

- **New Task Management** – This business process communicates with an external production control system, which is considered in this study to correspond to a non-CIMOSA-conformant domain. The "new task management" business process retrieves new orders and confirms the capability of the cell to accept them.

- **Task Allocation** – This business process classifies and schedules tasks and assigns them to
resources. It also dispatches specific tasks to particular resources at specific times (i.e. according to a generated work-plan). During the dispatching of tasks, different service tasks may be spawned from the original task, such as "providing tools" and "transferring materials".

The **Operation Management** domain process was decomposed into the following *business processes*:

- **Group Task** – This *business process* classifies tasks for each cell group.

- **Operate Specific Task** – This *business process* prepares necessary data and resources so that they can operate on single jobs related to each task and generate messages as required.

The **Monitor Cell** domain process was decomposed into the following *business processes*:

- **Display Specific Enquiry** – This *business process* is responsible for defining how specific data is displayed and recorded in response to particular queries about cell items, including the status of tasks, jobs, operators, resources, orders, etc.

- **On-line Monitoring** – This *business process* provides run-time information about each element in the cell, over some interval of time (as determined by capabilities of system hardware and software).

- **Analyse Data/Generate Reports** – This *business process* will become activate following either a "Display Enquiry" or "On-line Monitoring" *business process*, and will function to analyse results and provide appropriate reports. Evaluator tools that may be assigned to a cell, like simulators or emulators, could support performance analysis in this **domain process**.

In accordance with the CIMOSA approach, each *business process* comprises a set of related **enterprise activities**. The **enterprise activities** were specified by using individual templates provided by the SEWOSA tool. These templates organise the objectives, links and *object views* (e.g. information and resource objects) associated to each **enterprise activity**.

**Enterprise activities** and *business processes* will operate in a logical order as defined by *procedural rules*, thereby modelling the behaviour of real cells. These rules can be enacted based on the defined *End Status* of the **enterprise activity**, i.e. after termination of particular operations. Different types of *procedural rules* have been defined by the CIMOSA specifications. Figure 5.6 illustrates a sample operational procedure for a **domain process**, modelled by the 'behaviour diagram' of the SEWOSA tool-kit. The template accompanying this diagram defines the **procedural rules** related to the **domain process** under consideration. Behaviour diagrams can be analysed at each level of decomposition. Further decompositions may be modelled using child behaviour diagrams. The reader can refer to Appendix F for the complete decomposition of ‘behaviour diagrams’ and **enterprise activities** defined in this study to define the conceptual design of the semi-generic model of manufacturing cell system.
The manner in which semi-generic enterprise activities will be executed was defined as some logically ordered sets of rules specified using “Reference Procedural Rules” (RPR). These rule-based descriptions were coded using a textural format to be input to functional models. Thereby they specify execution flow at a semi-generic level. The “reference procedural rules” will be used as part of the design specifications, later at the “detailed design” level of semi-generic model. A sample set of “reference procedural rules” is shown in Appendix B.

Figure 5.7 illustrates the functional diagram of the new task management domain process generated using the SEWOSA tool-kit. Inputs and outputs to each enterprise activity have been defined as an information object view, a resource object view, and a set of events inherited from the domain diagram.

For example, the activation of enterprise activity EA-1 (receive new order) follows the combined occurrence of two events; namely control inputs, and information object views (e.g. OV-2, order information). Then it generates information OV-14 (e.g. suitable message).

To fully define a CIMOSA model, the input/output resource objects must also be specified. However, the means of presenting resource objects were not considered to be covered in the semi-generic model as it was outside the scope of the research. When developing a “TO-BE” model of a
semi-generic manufacturing cell control, the actual resources can be left undefined. However, these resources need to be defined later at the particular requirements definition level of modelling. The implication of such a restriction on the research scope will be discussed later in this thesis.

It should be re-emphasised that the above definitions of semi-generic tasks and their corresponding functional decompositions are built by selecting and assigning relationships between a set of sample tasks, previously specified for a restricted domain of manufacturing cell. It is not presumed that good coverage is provided by the sample tasks, rather they have been defined to test the concepts being investigated in this research. For different domains these generic tasks may be modified (simplified or complemented by new task definitions) in order to meet identified requirements of new manufacturing domains.

The actual functionality of each enterprise activity must subsequently be analysed at the design specification level of modelling, as a set of Function Entities (supported by Function Operations). However, currently the CIMOSA specification does not propose a clear transition process from a requirement definition model to models developed at detailed design levels, as mentioned by Aguiar [117] and stated by AMICE [56]:

“A clear definition [of links] between requirements level and design level is difficult to establish precisely and is still to be stated in CIMOSA. How complete a requirements definition model should be in terms of functionality is still an open question.”

5.3.2) Modelling Information Aspects

The ability to manage information is an important ingredient in any effort aimed at co-ordinating control activities in a complex organisation [149]. It is necessary to understand information
requirements that should be captured to sufficiently represent a process [95, 150]. Therefore, as part of a semi-generic approach to the modelling of manufacturing cells, it is necessary to specify the information requirements as well as functional requirements.

To effectively implement the information systems in an enterprise, "generic information entities" need to be modelled in a stepwise manner in parallel with the modelling of generic functional requirements (as outlined by Table 5.1). An enterprise depends heavily on ability to handle and use information efficiently and effectively and to implement necessary operational modifications in real time [62].

Therefore, it was considered to be a key requirement of this study to model information and function viewpoints of cells in a coherent way. Hence, this research considered three stages at which a supportive information system should be specified. It was assumed that the process of modelling enterprise information should begin with defining information requirements at a high level of modelling (i.e. at a semi-generic level). At the next stage, formalising and use of information should be consistent with data structuring defined at the first stage. Hence an information (and data) model\(^5\) was generated during the system design to operationalize the use of information model during system execution. At the last stage, the information system\(^5\) should be able to underpin a connectivity between and integration of information objects (generated at high level modelling, i.e. conceptual design level) with the physical data and resources (e.g. data storage/data exchange devices) required to realise system operation.

In this section of the thesis, the semi-generic and conceptual information requirements of manufacturing cells is discussed in accordance with the information view defined by the CIMOSA specification. Formalisation of the information requirements should be completed by defining suitable information models and characteristics of associated information systems. The later concern will be discussed in chapters 6 and 7. The semi-generic information models developed in this research has been also reported in a separate document by the author [151].

5.3.2.1) Information View of the Semi-Generic Model

To improve acceptability of its findings, the information modelling approach developed in this study conforms to the CIMOSA information modelling approach as far as possible. Thereby, it may prove practical for resultant domain knowledge (i.e. in the form of the semi-generic model of cells) to be reused in various manufacturing domains.

Hence in this context it is important to describe in some further detail the information modelling concepts defined by the CIMOSA consortium. According to Jorysz and Vernadat [152] the

\(^5\) Information modelling in this research is considered to be a combination of concepts and methods that design and formalise the reuse of information objects within manufacturing systems.
information view of CIMOSA consists of: (A) a frame for information modelling, based on an Object-entity Relationship Attribute approach; (B) the three schema approach, as defined by ANSI; (C) information modelling, as an integral part of the CIMOSA life cycle; and (D) a representation formalism. Figure 5.8 (left side) illustrates the information view of the CIMOSA framework at its various levels of modelling.

During requirement definition modelling, information models are created using the following CIMOSA modelling constructs: Enterprise Objects, Object Views, Information Elements; and an Object Abstraction Mechanism [17]. An object view is a description of a particular aspect of an enterprise object. It is made up of information elements and other object views. An information element is an indivisible piece of information and corresponds to the atomic level of data modelling in CIMOSA. Enterprise objects are information entities that form part of the enterprise and are defined by attributes, that can be either information elements or enterprise objects modelled in greater detail at another level in an object hierarchy. The objects are linked together by means of an object abstraction mechanism.

Figure 5.8: The CIMOSA information model and semi-generic information model developed in this research (left part of the figure was adapted from [152])

---

6 An information system (and information services [56]) in this research is considered to comprise methods, techniques and tools that enable the system to manipulate data including: storage systems; data exchange methods; data update issues; connectivity issues; and data formatting.
5.3.2.2) Semi-Generic Information Objects

During process-oriented modelling (leading to a requirement definition), enterprise activities and their relationships were defined in the manner described in §5.3.1. The inputs and outputs of an enterprise activity are sets of object views (see figure 5.7). These object views can therefore be related to enterprise objects and can be Physical Objects or other (modelled) Information Objects.

At this level of enterprise modelling the information view of the semi-generic model of cells was developed by defining a set of common information objects that support common information requirements of cells used in the PCB manufacturing domain. Subsequently different populations of these objects (by inputting specific data) can provide different instantiations of the semi-generic model, i.e. a particular model of a target cell and its manufacturing applications.

Table 5.2 depicts a sample set of generic "information entities" defined in this way. This includes: enterprise objects; information object views and information elements that form part of the semi-generic information model. These information entities should be populated with the real data pertaining to an actual cell. The information entities may also be removed or modified based on the information requirements in an alternative domain. A complete definition of the information entities modelled in this study is included in Appendix C.

<table>
<thead>
<tr>
<th>Enterprise Object</th>
<th>Information Object Views</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordering</td>
<td>New Orders List</td>
<td>Order list from out of cell domain</td>
</tr>
<tr>
<td>Task management</td>
<td>Main Tasks List, Transporting Tasks List, Material List, Tool Kit, Auxiliary Equipment</td>
<td>Main tasks generated by scheduler, optional transporting tasks, necessary tools material and equipment for main tasks</td>
</tr>
<tr>
<td>Messaging</td>
<td>Message</td>
<td>optional message type (e.g. error, command, emergency, normal, etc.)</td>
</tr>
<tr>
<td>Supporting Data</td>
<td>Eng. Data, Sys. Static Data, Device Data, Temporary Not-Working Configuration Data</td>
<td>pre-define Eng. data for orders and cell items specification, data saved in database, items, not be included in scheduler cell networking configuration data</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Dynamic Data, History Data, Formatted Data, Report</td>
<td>run-time data from variable items, result of operations, associated by static data, associated by history data</td>
</tr>
</tbody>
</table>

Table 5.2: Some of the semi-generic information entities defined at the requirement definition modelling level

The information entities defined at the requirement definition level should be capable of transfer in a consistent way to the design level of modelling. These entities should be organised in a logical format so that a computer can process them. Ways in which these entities can be organised will be discussed in chapter 6.
Summary

This chapter discussed the approach adopted when designing the semi-generic model of manufacturing cells. It explained that the semi-generic model comprises a "conceptual design" level and a "detailed design" level. At the conceptual design level of semi-generic modelling, the functional requirements of cell systems are specified. These functional requirements included: a) formally structured generic activities; and b) generic process flows of those activities.

Commonly information requirements of a given manufacturing domain were also specified at this level of semi-generic modelling.

The functional aspects of the semi-generic model were developed in conformance with the CIMOSA modelling approach, and were captured and applied by using the SEWOSA CASE Tool developed previously by researchers in the MSI Research Institute.

The information aspects of the semi-generic model were also specified based on an application of CIMOSA principles to the modelling of information views. Thereby, generic information entities were formally defined (in a textural format) and used to structure use of a standard information modelling method based on use of the EXPRESS modelling language at the detailed design level of the semi-generic modelling.

The next chapter of the thesis will discuss how the development of semi-generic model was continued at the detailed design level of modelling.
6.0) Introduction

Early chapters of this thesis have identified the conceptual requirement for the development of change capable manufacturing cells, capable of being rapidly configured to accommodate changes in their manufacturing environment. It was also concluded that a modelling approach was required to support the formal design and implementation of particular cells and cell control systems. To facilitate the life cycle engineering of cells, the use of a particular model of a cell and its control systems was proposed. Hence, when system specifications change, a higher-level of modelling structure (i.e. a semi-generic model of cell and its cell control systems) was proposed to formally structure and support the modification, development, and reuse of particular models. The development of such semi-generic structure at the conceptual design level of modelling was discussed in chapter 5. This led to the definition of a sample set of generic and reusable system entities (including functional and information entities) for given domain of manufacturing cell systems.

This chapter is concerned with extending the design of semi-generic model to the detailed design level of modelling (see research plan in figure 4.2). Here, focus is on design and implementation of a component-based manufacturing cell control (MCC) system, as it was envisaged that a change capability could be embedded into specific implementation of target cell by being able to readily reconfigure the control system.

During the detailed design of the semi-generic model, the generic control system entities should be flexibly linked to physical cell resources. Hence representation of these entities would also need to be included in the semi-generic model at the detailed design level.

Discussion in this chapter is centred on:

- Defining modelling entities to represent real-world physical resources,
- Designing a semi-generic information model to support detailed design of cells and cell control system,
- Defining design specifications related to the semi-generic model using a component-based approach.

1 It was realised that improving reconfigurability of physical resources in a cell (i.e. machines, devices, etc.) can contribute significantly in the overall configurability of cells and their cell control systems. However, this research places less focus on this aspect of configurability of cells.
6.1) Detailed Design of the Semi-Generic Model

At the conceptual design level of semi-generic modelling, the requirements definition of the modelling structure was specified in form of the "generic elements" and their relationships (§5.1). It was understood that the modelling structure should be extended to the "detailed design" level of modelling to include the design and implementation of component-based manufacturing cells controllers. Naturally it was assumed that any extension should be consistent with that of the conceptual design part of the semi-generic model.

Early experimental work in this research study investigated the use of two different approaches to designing the semi-generic model. An initial model was developed and structured by using the SEWOSA tool. A description of this initial model is given in Appendix E. Here it was found that the SEWOSA tool offered a well-defined design approach applicable to the general modelling of manufacturing enterprises. However, by being based wholly on the CIMOSA specification, naturally the SEWOSA modelling constructs do not target system implementation towards any specific architectural style (e.g. a component-based solution) or use of a specific set of software engineering tools. In addition, SEWOSA was not explicitly constructed with the design of component-based systems in mind, rather its implementation focus was targeted on the development and execution of use of CIMOSA business entity models that operate within runtime environments to synchronise and control the interoperation of runtime system elements. Therefore, after analysing the results, a number of constraints were identified in respect of a pure use of the SEWOSA (and hence CIMOSA) approach when using it with the purpose of engineering change capable manufacturing cells (see the discussion in Appendix E).

Consequent on this initial exploratory study the second approach was designed and developed which specifically targets modelling towards the use of component-based concepts. As explained in the remainder of this thesis, the second approach was found to have various advantages relative to the earlier approach.

Although the CIMOSA modelling structure was not found sufficiently suitable for the detailed design part of the semi-generic model, it does include very useful primitive modelling constructs termed Functional Entities (FE's) and Functional Operations (FO's) that it was understood might be used in two main respects, namely (I) to maintain connectivity and consistency between models of cell requirements and models of cell control systems and cell resources and (II) as primitive modelling constructs that might have the capability to represent models of reusable cell control system components and resource elements in cells. Here it was assumed that a key general requirement of an effective approach to the life cycle engineering of component-based manufacturing systems would be a capability to maintain sufficiently accurate and complete links (or mappings) between models of generic elements and real physical elements of manufacturing cells, i.e. it would be necessary to achieve a sufficient connectivity between the modelling
environment, configured cell controllers, and physical system resources.

It was also understood that it would be necessary to model information elements utilised during detailed design, implementation and runtime operation of cell controllers. Furthermore these models would need to be consistent with and enable 'connectivity' to information models used to define wider-scope manufacturing cell requirements and conceptual designs. In this respect it was assumed that the published CIMOSA specification would include sufficient modelling constructs. However CIMOSA did not specify use of any particular information modelling standard. Therefore it was assumed that it would be necessary to select use of such standards and where necessary to provide information modelling tool support to test the project concepts. For reasons stated earlier STEP and EXPRESS information modelling concepts were adopted in this case to complement the use of CIMOSA modelling constructs.

Figure 6.1 illustrates the detailed design part of the semi-generic model of manufacturing cells and the approaches deployed at this stage of the research. Therefore, an enterprise modelling approach (EM) was employed to formalise the operational elements of the modelling system (i.e. BE’s and FE’s). Whereas an enterprise integration approach (EI) was used to define and establish communication methods capable of flexibly integrating modelling elements and real physical resources. A standard information modelling method was also implemented to formalise the information required in a consistent way during modelling and real-time execution of target systems. In addition, the component-based design approach (CB) was deployed to assist in the engineering of change capable manufacturing cell controllers.

6.2) Designing Functional Aspects

Semi-generic tasks (represented as sets of EA’s and relationships between EA’s at the requirement
definition level of the CIMOSA modelling structure) need to be related to organised descriptions of Functional Operations (FO's). In the CIMOSA modelling structure FO's are atomic representational building blocks of processes at the design specification level. The CIMOSA modelling approach requires FO's to be treated as computer executable messages that are assigned to operational elements of a CIMOSA model. These CIMOSA operational elements are units of functionality, termed Functional Entities (FE's). A FE is a generic form of some active elemental building block of the system. In general, these will be human, machine and software elements. Normally FE's will be organised into larger grained active function blocks of a system. It follows that CIMOSA methods utilise FE modelling constructs to represent defined classes of physical resources and generally these definitions will form part of a resource modelling view of CIMOSA enterprise models. Hence a first step taken in this study to create the second part of the semi-generic model of component-based cell controllers (i.e. detailed design level) was to specify a set of generic function entities for the domain of PCB manufacturing cells. Here the following set of function entities were specified, as a basic set that could test the component-based systems engineering concepts proposed and developed as part of this study:

- **Ordering System function entity** - The purpose of this FE is to receive new orders from external entities outside the scope of the manufacturing cell being modelled. Typically these entities might be a MRPII package or a manual interface providing a human-centred means of generating and issuing new orders.

- **Cell Supervisor (human) function entity** - This represents a human supervisor of the cell supported by a “cell supervisor” application tool. Typically a supervisor will be responsible for co-ordinating activities like the receiving of new orders, inputting data to and interpreting results from a scheduler/dispatcher software package and requesting specific items concerning the cell status. In general interactions between the supervisor, supervisory tool and manufacturing cell will be realised by sending and receiving messages. The supervisor must initiate courses of action in the event of errors, breakdowns and external requests. Also the supervisor must interpret requirements from and feed back reports to higher levels of control. Here it was assumed that communication with other parts of the manufacturing system (i.e. with other CIMOSA and non-CIMOSA compliant domains) would be established under direct supervision of this function entity.

- In practice it was understood that the responsibilities of this function entity may vary considerably and be highly dependent on the level of automation deployed within a cell. Therefore, the structure of the semi-generic model of cell controllers was designed to unify the engineering of semi-automated cells. Hence, many activities in the semi-generic cells are assumed to be operated automatically after confirmation by human operators.

- **Cell Operator (human) function entity** - This FE represents a person responsible for various
tasks associated with the run-time operation of the cell, such as setting up machines and devices, loading and unloading of materials, and confirmation of the completion of tasks or sub-tasks. It was assumed that this function entity would operate in conjunction with Operator/Group Application function entities. The activities and responsibility assumed for any given Cell Operator FE were expected to depend on the level of automation deployed within a given cell.

- **Scheduler/Dispatcher function entity** - This class of FE is required to provide the functional capabilities of a scheduling software package (or an interface to an external scheduler system) that operates within the scope of a modelled cell. Hence it should be capable of generating work-plans for new tasks, generating descriptions for supporting tasks (such as material handling operations) and despatching tasks to actual resources. This function entity should provide schedule lists for supporting equipment including tools, materials (raw material and semi-finished parts) physical resources (machines, operators) and other auxiliary equipment (e.g. jigs, fixtures).

- **Cell Supervisor Application function entity** - This is a set of computer programs that via a user interface provide computer support for the cell supervisor. It was assumed that this class of function entity would facilitate access to external manufacturing systems and tools, such as a MRP II package, a stock control system, and/or a maintenance system. It was also envisaged that, this function entity would function co-operatively with supporting systems such as a local database system or an external data storage system (serving a broader scope shop-floor of factory), as well as internal entities of the cell such as human operators, software application tools used by specific operators, a monitoring system and so on.

- **Operator/Group Application function entity** - This class of FE corresponds to a set of computer programs that provide user interface functions to control and support activities carried out by different operators working in a cell. Because it was assumed that typically a cell would comprise a number of groupings of largely independent resources and associated activities, it was also assumed that this class of FE should act as an interface between the cell and each grouping within it. It was also assumed that group applications should be co-ordinated by the group supervisor application. Based on this assumption it was determined that the responsibilities of this function entity should be as follows: receive allocated tasks from the cell supervisor; retrieve appropriate engineering data for each task, by accessing a local (or remote) data storage system; provide job lists for cell tasks (it being assumed that tasks will consists of set of jobs); control execution of each job by using the main resource controller and connectivity to appropriate resources (via function operations); remove the jobs (or tasks) from associated queues after completion of jobs and create a history file. It was assumed that the number of groups and operators in each cell would determine the number of this class of application. Hence, each application should be configured for particular implementation in
different cells.

- **Physical Devices function entity** - This class of FE models the actual physical devices in a cell, or an interface or driver to such devices. Therefore the number of these function entities will be depend on the number of physical devices (or drivers to the devices) in the cell. Examples of these devices include a machining centre, a PCB test machine, an assembly robot an AGV machine or an automatic palette changer. This function entity should be enabled to issue and receive, interpret standard formatted messages exchanged with the actual physical devices installed into a cell.

- **Message Dialogue function entity** - This is a software application that handles the exchange of all messages within the scope of the cell. It was assumed that this class of FE should receive different types of message from specified FE’s within the cell, categorise the messages into various types, and send them to defined destinations. Such a software application was expected to function as a part of cell supervisory application. It was also expected that the efficiency of this class of FE would be highly dependent on properties of the integration infrastructure used by the system as a whole.

- **Monitoring function entity** - This class of function entity was designed to collect: a) dynamic data associated with defined FE’s of a cell; and b) static system data including for example physical system configuration data and engineering data for each FE (see Appendix C). It was assumed that this function entity should provide sufficient data to the run-time control system to enable it to generate a specific cell status report on demand. It was envisaged that it should also deal with aspects of data analysis and generate overall reports about the performance of the cell.

Having defined the functional elements of cells into the above FE’s classes, the research study was focused on more formally defining semi-generic enterprise activities and function entities, as well as specifying links between these elements (i.e. functional operations) and links with supportive information system elements.

### 6.3) Designing Information Aspects

Common information requirements of PCB manufacturing cells were specified when developing the first part of the semi-generic model of cells. These definitions were defined in the form of “generic information objects” (§5.3.2.2). At the detailed design level of semi-generic modelling (focused on developing a semi-generic model of a specific component-based cell controller) the “generic information objects” must be attributed to “functional elements” of the modelling structure. However it was understood that this attribution should be dependent on the methods selected to perform cell tasks and also on properties of the resources assigned for use in the physical cell. Therefore these information objects cannot be completely defined during requirements modelling of cells. Rather at this stage they must be in an abstract and incomplete
form. However it was assumed that it would be possible to define a useful set of information objects capable of encoding general requirements and properties of cells that can be used to guide the design, engineering and change of component-based cell control systems. Hence the information objects specified in respect to the first part of the semi-generic model were categorised into different sub-groups, where classification was based on their relationship to generic tasks.

However to facilitate use of the generic information objects during the design, construction and change of component-based cell controllers it was necessary to develop, as part of this study, a supporting software tool. When selecting an information modelling approach particular concerns were (a) a need where possible to conform to CIMOSA information modelling concepts to facilitate reuse of the resultant domain knowledge; (b) to deploy standard information modelling methods and techniques, to facilitate integration with other systems, typically operating in a host manufacturing environment. To achieve these aims various relevant object-oriented paradigms were reviewed (see literature review §2.4) and their capabilities were assessed with respect to the requirements of this research.

At the design specification level of CIMOSA modelling, use of a standard information architecture based on three-Schema approach (i.e. conceptual, internal and external schema) was selected, specified and developed. This approach structures the transformation of object-oriented information descriptions (obtained at the requirement definition modelling level) into an equivalent data-oriented information model [152]. On considering the practicality issues related to use of the CIMOSA information modelling framework, a decision was made to adopt its use to guide object-oriented modelling during the detailed design part of the semi-generic model. According to CIMOSA, at the requirement definition level, information objects must be defined in a descriptive format. Therefore it was decided that a formal method should be used to enable the information objects to be transformed into a computer-processable format. Hence, it was necessary to choose specific information modelling formalisms to design and produce tools to populate the design specification level of CIMOSA modelling, as the CIMOSA Consortium had not prescriptively specified enterprise engineering requirements in this respect [56, 153]. After review, the author chose to adopt the modelling formalisms advanced as part of the STEP standard [101] and EXPRESS language [99]. The rational for choosing these methods was discussed previously in section §2.7.

Objects defined in the EXPRESS language benefit from the structure and inheritance capabilities provided by this information modelling language. Use of an EXPRESS model was proposed to define the conceptual schema of the semi-generic information model, based on the use of a three-schema approach. By so doing, data classifications embodied in the model would naturally be organised within a structured hierarchy. Therefore, each data class would inherit attributes from one or more super-class of data in that hierarchy. Also information elements would be properties of a data class that can be shared by different information objects. Furthermore, it was envisaged that
by allocating different values to each property of the semi-generic information model it would prove to be practical to spawn different specific information models, each can be used to describe information aspects of a particular manufacturing cell controller.

It was also realised that when implementing any system that deploys centralised database support, the information objects could structure the tables in a physical database. Whereas the values of the information objects would be the actual data stored in the tables [17]. It was also decided that inputs to and outputs from the EXPRESS model (via associated database transactions) should take the form of ‘STEP physical files’ [100]. To achieve this aim as part of this study it was necessary to establish an approach to mapping CIMOSA information descriptions onto constructs of the EXPRESS language. The following section describes the approach developed for this purpose.

6.3.1) CIMOSA-EXPRESS Mapping

Transformation of the CIMOSA information objects into an equivalent representation using the EXPRESS language requires the definition of an equivalent EXPRESS model construct for each class of information entity defined in the CIMOSA model. EXPRESS 'schemata' (that are at the highest level of EXPRESS modelling) need to be structured by properties of the enterprise information object. 'Entities' in the EXPRESS model correspond to Information Object Views in the CIMOSA model, hence these 'entities' can have relationships with other 'entities' within the 'schema' and maintain loosely defined relationships with 'entities' in other schemata of the EXPRESS model.

Furthermore the 'entities' should have 'attributes' that may have different types and values. In this study the 'attributes' need to be built from CIMOSA information elements. Like information elements, 'entity attributes' can be pieces of data of different types (i.e., integer, string and so on) or a set of data in the form of a set, or array, etc. Thereby, 'attributes' can be shared by different 'entities' or inherited from other 'entities'.

Although the need to build EXPRESS models that include several external 'schemata' may complicate the practical application of an information modelling system and also increase the complexity involved in developing modelling tools, it was assumed that it would provide sufficient means to cope with the high levels of complexity inherent in the domain of manufacturing cell systems (e.g. the target application domain) demands such formal approach. Previous research studies have shown that it is necessary to develop well-structured models that can be changed readily by system designers, builders, and developers (from their different foci of coherent), whilst retaining an inherent ability to select, delete, update, reuse, scale up, abstract, etc. information models.

Table 6.1 lists the different classes of the information model identified and formalised as part of this study to support the requirement definition and design specification levels of CIMOSA modelling.
6.3.2) Design of a Semi-Generic Information Model

The first part of the semi-generic model of manufacturing cells developed in this study included definitions of enterprise objects, information object views and information element (see Appendix C) that can be used to structure and facilitate the requirements definition of particular PCB cells. However information entities need to be instantiated and assigned to specific cell entities during the design of cell controllers. Furthermore, for reasons explained above, it was decided that this should be formalised by utilising EXPRESS modelling constructs. It was also decided that the instantiation and assignment of life cycle engineering activities should be done semi-automatically by supporting designers, builders and developers of cell controllers with the use of an appropriate interface that facilitates access to knowledge of the mapping technique\(^2\). Based on this approach sets of CIMOSA-conformant information entities were redefined using the EXPRESS modelling language. Table 6.2 indicates the 'schemata' and 'entities' defined that correspond to the information structure defined earlier via the CIMOSA modelling. As discussed earlier, the 'schema', 'entities', and 'attributes' respectively correspond to enterprise objects, object views, and information elements, defined at the conceptual design level of the semi-generic model.

\(^2\) Initially in this research, these conversion processes were carried out manually. Later, a computer programme was produced to assist the conversion (see Appendix D). However, it was not recognised as a priority task of this research to generate a graphical user-friendly tool to support the conversion process.
Figure 6.2: EXPRESS-G representation of the semi-generic information model at the schema level (for complete illustrations see Appendix C)

Figure 6.2 illustrates a part of the EXPRESS model and its graphical representation in EXPRESS-G\(^3\) generated in this way. A complete graphical and textural representation of the EXPRESS schema at the 'entity level' is included in Appendix C.

6.4) Further Development of the Component-Based Approach

The foregoing sub-sections of this chapter illustrate how semi-generic function and information requirements of the manufacturing cells can be formally mapped onto semi-generic models of functional entities and information entities of cell control systems. This was achieved by defining FE’s and operationalizing the use of CIMOSA modelling constructs at the detailed design level of semi-generic modelling. It also required mappings to be defined and supported that link CIMOSA modelling and EXPRESS language constructs. However, to develop a component-based design approach it is also necessary to define relationships and interaction methods used by generic cell control elements, physical resources to be used, and also other entities outside the system model scope (such as non-CIMOSA domain processes - §5.3.1) and external supporting tools (e.g. MRP II, Quality and Material Tracking systems).

As explained previously, up to this stage research effort was focused largely on the use of CIMOSA-conformant approaches. Therefore the semi-generic model was initially constructed and tested using the SEWOSA tool [117]. Based on the results concluded when deployed this approach (see Appendix E), it was found that CIMOSA (and SEWOSA, which supports the use of function

---

\(^3\) Other MSI researchers had developed an EXPRESS-G tool using the IPSYS CASE tool. This generates EXPRESS language descriptions from EXPRESS-G constructs and models. The tool can cover most aspects of the EXPRESS language, hence the tool was used in the development of the semi-generic information model. A further proprietary software tool that used to check the EXPRESS syntax of the model text, was 'ICE version 2.0' [154].
oriented CIMOSA concepts) has certain limitations, particularly with respect to the way that models are operationalized in support of the subsequent implementation, runtime and change of systems. Indeed to address certain of these shortcomings other researchers have investigated ways of associating various standard (or well established) modelling methods, techniques and tools to the CIMOSA modelling approach. Example methods and techniques assessed in detail with a view to unifying their use with CIMOSA concepts included: EXPRESS and ERA [155], mainly with a view to formally modelling the CIMOSA information view, and Estelle [156] and Petri-Nets [157] with the purpose of formally modelling the CIMOSA functional view.

Therefore in this research, alternative approaches were investigated particularly with respect to the use of more formal software engineering concepts that promised to enable computer executable models of cell controllers to be designed in detail and used to help structure and support other life phases of cells, such as their construction, runtime, operation and management and reconfiguration and ongoing development.

It should be pointed out however that the CIMOSA architecture was designed with integration to other methods in mind, with an aim to provide comprehensive large-scale systems engineering support. Therefore on considering further extension of formal modelling capabilities for component-based cell controllers (e.g. to facilitate function entity relationship and component interaction and behaviour modelling) a key technical consideration in the choice and development of any modelling and software engineering paradigm was that its use should be consistent with that of CIMOSA modelling constructs used during process oriented modelling to capture cell requirements. Also it was assumed that choice should not impair reusability and generality characteristics that should be inherent properties of any successful ‘component-based’ approach (§2.6). Therefore an approach based on use of a backbone of CIMOSA concepts should facilitate the specification and aggregation of well proven and reusable cell components and their interoperation in alignment with defined system behaviour [28, 125]. Implicitly therefore, it was assumed that such an approach should offer the following advantages.

- it should naturally support a formal decomposition of manufacturing cells and their controllers into reusable objects (i.e. well defined classes) of ideal control system component and resource element that can be configured and can interoperate in a variety of ways.
- it should generate computer executable models of objects (or software components) that can be transformed into simulation and emulation models that support system behaviour analysis, system design analysis, system implementation and debugging, runtime operation and system design analysis and incremental change.
- it should enable the distribution and reconfiguration of system functionality via the use of “standard” object oriented integration mechanisms that enable interactions between well defined objects to be realised in a flexible way.
it should lead to modular modelling and engineering of cell systems and thereby rapid engineering and reengineering of medium to large scale systems in a cost effective manner.

Therefore an aim (when developing the approach proposed in this study) was to have the potential to impact on current practice when engineering and configuring systems, and more particularly in the context of this study when designing and building manufacturing cells from reusable components. Indeed potentially such an approach can facilitate model driven operation, model driven configuration and real-time data capture and processing in manufacturing cells [118]. However, the practicality of this kind of approach had to be assessed to determine the nature of the real-world constraints that might mitigate against its adoption in the field. Hence a prime aim of this study was to develop a component-based approach in proof-of-concept form focusing on issues that were considered to be of prime concern in the domain of manufacturing cells. Necessarily the approach needed to be focused and pragmatic to some extent. In this context Figure 6.3 illustrates the coverage of the proof-of-concept component-based approach.

Essentially, as illustrated in chapter 5 and early sections of this chapter, two main decomposition approaches were developed and utilised in this research. One thread decomposition breaks down abstract models of business and production requirements into less abstract models of ‘tasks’ that are expressed in terms of enterprise modelling constructs (e.g. EA’s) that can be transferred into executable modelling constructs (§5.2). Whereas the second thread concerns the breakdown of known cell solutions and design methods into models of reusable components (e.g. typical resources used in PCB cells) and groupings of those components to represent models of alternative candidate systems [18]. Both enterprise modelling constructs (e.g. FE’s) and formal information modelling elements (i.e. EXPRESS constructs) are part of the second approach. These modelling formalisms (and associated software engineering constructs) were used in this research to specify three types of component. These include:

- “Reusable Activity Components”. These components take the form of generic tasks or enterprise activities. They may comprise a single component or an aggregation of several
components. An “activity component” specifies what a system should be capable of doing. This capability should be matched to actual system requirements defined at the requirements definition level of system modelling. Also if process change results in a new system requirements definition then appropriate change should be made to the selection and/or configuration of “activity components”.

- “Reusable Resource Components”. These components are required to realise well-defined units of functionality needed to (a) realise tasks (and therefore the elemental activities) previously defined and (b) organise (i.e. synchronise, sequence and monitor) the interoperation of the units of functionality selected so that they realise tasks (and elemental activities) previously defined in the right order and on time. Therefore configured sets of functional components (comprising resource elements and control system components) need to be specified at the system design level of modelling.
- “Reusable Information Components”. These components need to support the interoperation of the functional components (i.e. resource entities) and control system components with information objects that can be flexibly attributed to both activity and function components.

6.5) Resource Allocation and Configuration Method (RACM)

The experience of developing a semi-generic model of manufacturing cells and the cell controllers led in this study to the specification and development of a component-based approach to the design of manufacturing cells, which was designated as the “Resource Allocation and Configuration Method” (RACM).

The principal concepts that underpin the RACM approach are depicted in Figure 6.4. In the context of this thesis RACM can be defined as follows:

"allocation of predefined and generic modelling entities and associated physical elements (called system components) to application capabilities (called software components) which enables performing a specific class of tasks in a manufacturing cell".

As discussed in the forgoing (§5.2), practical experience gained as part of this study led to the notion that significant benefit can arise if a manufacturing cell can be decomposed into large grained “functional groupings”, where each group is responsible for realising a sub-group of essentially stand-alone tasks, that can be formally defined as a common class of business entities.

The RACM approach was developed on the assumption that it would prove practical to “wrap up” each generic task into a software component (see figure 6.4) in such a way that it can be treated as a “reusable activity”. If such a capability can be realised (during the detailed design level of modelling), a system designer can match configurations of reusable components to specific sub-groupings of tasks associated with a “cell group” (see §5.2) in manufacturing cells. Implicitly in the context of this study the software wrappings had to be designed to be conformant with CIMOSA and software engineering concepts developed to realise the semi-generic modelling
structure. On following this line of thinking, components defined based on the RACM approach included the following entities.

- **Business Entities** – These are modelled elements that are reused during requirements specification. They comprise (1) generic tasks logically structured in the form of DP’s, BP’s and EA’s, (2) generic information objects and (3) procedural rules that define the flow of actions and events within the domain of manufacturing cells.

- **Resource Entities** – These are modelled elements that represent a generic class of physical resource (including human resources, machines, and supporting tools) with a functional capability to realise enterprise activities and therefore tasks. These modelled elements can be reused during system design.

- **Software Components** – These are modular, reusable and executable models of generic tasks (represented in form of software applications) that can be re-deployed in the manufacturing cell domain. Because they are executable models they can be manipulated in the design environment as well as being executed as part of an operational manufacturing cell. Hence their reuse can be consistent with that of business and resource entities and can operationalize their concepts, thereby providing mechanisms that cross the boundary between modelling environments and actual cell control environments. In the semi-generic modelling environments used in this study the software components are represented, configured and reused in the form of “templates”. While in the particular target control system environment they take the form of software applications, (e.g. function code and applets) that interoperate with other software components, machines and humans to realise cell tasks.

Ideally it was assumed that it would be appropriate to wrap tasks at the EA level of granularity, as this would facilitate extreme levels of flexibility when configuring cells. However, because of project constraints in this proof-of-concept study it was decided that it would prove practical

---

4 It is assumed that most system activities will require involvement to some degree of human resources (e.g. via supervisory or direct actions). However, the proof-of-concept application tools generated in this study are only able to technically support the automated (unmanned) operations by removing the human resource roles from the processes.
and sufficient to investigate ways of wrapping and reusing tasks at the business process (BP) level of granularity. Therefore, for example, business domain process 1 (see Appendix F) would be represented and reused as a software component that enables the cell supervisor to interact with software of an external scheduler.

- **Models of Cell Groups** - It was decided to realise a facility to reuse models of cell groups in a modelling environment. These models specify the definition of how manufacturing cells should be built from selected resource entities and software components. In this way descriptions of cell groups would also specify how responsibilities for realising defined tasks are sub-divided between groups.

- **Connectivity Objects** - This class of system component was defined with the purpose of interconnecting between implementation-independent software components (i.e. to each other and to resource entities) via implementation-dependent interfaces to the services of an integration infrastructure system which has appropriate integration capability to support cell operation. It was found to be difficult to design this type of object in a generic manner. However these problems were eased if a well-established underlying (and preferably standard) integration mechanism (e.g. OMG - CORBA [58]) was deployed by the system. In such cases the design of the software components should comply with requirements of these mechanisms, in terms of the interaction method and data exchange format required. In this research, connectivity objects were defined in correspondence with the particular level of modelling in form of "application drivers" and "CIMBIOSYS functions" (discussed further in Appendix D) which can be attached to the software components to facilitate flexible integration of these components to the other system elements.

Therefore the RACM approach assumes that *function entities* and *business entities* are allocated and configured into groups that perform designated tasks in a largely independent and concurrent manner. Accordingly it is necessary to (re)configure the execution flow of *enterprise activities* via the attachment of suitable *procedural rules* that will enable all tasks in the cell to be realised in a defined order of processing.

To support use of the RACM approach described above it was found to be necessary to develop an application tool to provide means of describing and allocating system components to cell groups as well as means of generating software components that are capable of interoperation in a target cell controller. A prototype version of such a tool was designed and constructed to facilitate proof-of-concept experiments to be carried out in this research. The concepts involved and the development of this tool are discussed in section 7.5.

At the detailed design level of system modelling (as part of the RACM approach) system designers, constructors and developers need to abstract system entities in various steps, leading to generation
and manipulation of "software components". Abstracted system entities should specify 'what' tasks need to be done in the cell, 'where' these tasks will be carried out, and 'how' the tasks can be realised (and performed) via use of an available set of resources elements. They also need to specify and implement an appropriate network configuration to provide the integration service required. Hence it is necessary to specify: resource availability; "grouping" within a cell; ways of meeting functional entity requirements; ways of meeting business entities requirements; establishing the relationships between system entities; generating the "allocated procedural rules" (from "reference procedural rules"); and infrastructure services and supportive systems. Figure 6.5 illustrates the concepts embedded into the RACM approach within the detailed design level of the semi-generic modelling.

Furthermore a prototype application tool, which will be referred to as "Cell Design Tool" (CDTool) was specified and developed in this research to structure and underpin the
implementation of the RACM approach. The design and development of this application tool will be discussed in chapter 7. The remaining sub-sections of this chapter illustrate in some detail, how cell designers and cell developers might design and construct semi-generic cell control systems by using the RACM approach (also see figure 6.5). Subsequently, cell controllers designed in this way can be developed into a particular instance of a cell controller suitable for use in a specific PCB cell.

6.5.1) Review Resource Availability

The cell designer should define whether a new cell is required or if an existing cell needs to be re-engineered. This will require a detailed review of the capabilities required to perform the tasks involved and the capabilities and capacities of resources that can be assigned to that cell, including: human resources, physical devices and supporting software tools. Typically human resources deployed in cells will include cell operators (who may utilise computer tools) and a cell supervisor (see figure 6.6). The designer should also specify business entities previously defined at the conceptual design level of modelling.

6.5.2) Dividing Cell into Independent Groups

To facilitate configurability and enable concurrent operation (and hence improve cell performance) often a cell should be decomposed into largely independent groupings of tasks, as well as resources that can realise those tasks (§5.2). Therefore the cell designer must determine appropriate “groups” and model elements and relationships between elements comprising each group. Choice of grouping will vary in different cells and may change with different conditions in any given cell. A cell must comprise at least one group including the cell supervisor. At this stage of the RACM approach the designer actually defines responsibilities for tasks that must be performed by the cell.

6.5.3) Meeting Functional Entities Requirements

The means (i.e. Function entities) by which each “group” in a cell can realise and perform the tasks, should also be allocated to appropriate group. At this stage the designer should allocate functional entities to groups by precisely defining:

- who is responsible for managing each task (i.e. whether it be a human operator, a system component, or combinations of these),
• which physical device in a cell group will actually carry out the tasks, and
• what supporting tools are required (e.g. database, scheduler, MRPII package).

If a new cell is to be developed a TO-BE model must be defined. Typically this will begin by allocating business entities (defining "what to do in groups") and then allocating required resources. However in cases where a cell already exists, the allocation process may start from the use of an AS-IS model of the cell and resource configuration and proceed in a more bottom-up way toward a re-allocation of existing resources. Figure 6.7 illustrates the allocation of business entities and functional entities to cell groups.

6.5.4) Meeting Business Entity Requirements

The system designer should also specify which activities need to be carried out by each group in cell, by allocating activities as business entities. As CIMOSA model construction was used as a basis for RACM, business entities comprise: domain processes (which in this study correspond to three domains defined at the requirements level of modelling - see §5.3.1); business processes; and enterprise activities. By allocating a set of EA’s to a group, a designer prepares the system in a way that will structure subsequent system engineering processes involved in the generation of software components (i.e. during system implementation) based on access to well-defined activity specifications for each group. For instance allocating BP-4 (i.e. Resource Preparation from DP-2) to group-2 should result in the system having a capability to carry out the designated function operations corresponding to the EA’s in BP-4 (see Appendix F) for detailed diagrams).

Implementation issues regarding how these EA’s handle real physical jobs in a specific cell will be discussed later in chapter 7.

Figure 6.7: Allocation of business and function entities to the cell groups
6.5.5) Establishing Relationships between Cell Entities

Having assigned resources and enterprise activities to each group in a cell, relationships amongst the entities in a group and also between different groups in a cell should also be determined. This can be achieved by allocating a set of EA’s to each group and specifying which human resource is responsible for performing each EA, what physical devices are required, and which supporting software tools need to be used. An EA may need to be shared between different resources available to a group or between different groups of a cell. Also a human resource may need to maintain relationships with more than one device or supporting tool. Figure 6.8 illustrates the resource assignment concept developed and applied in this study, by emphasising on the role of human resources.

6.5.6) Attribution of Procedural Rules

At the requirement definition level of the CIMOSA modelling structure, procedural rules are used to define the execution flow of EA’s and their end status (see Appendix G). In this research, a set of “reference procedural rules” were defined at the conceptual design level of semi-generic modelling. These rules can be used to specify generic flows of EA’s in a given manufacturing domain (see Appendix B). However, in practice for a specific industry case (represented by a particular level of model) only some of these generic EA’s will be needed to be included into the definition of a particular model of a cell, based on the “tasks” assigned to cell groups. Hence it is necessary to establish means by which a set of procedural rules can be attributed to EA’s assigned to any given group. The set of procedural rules should specify the flow of EA’s even if these are not to be carried out in the same order as that defined by the semi-generic model (see the ‘behaviour diagrams, illustrated in Appendix F). The set of procedural rules attributed to each group will be referred to as “Allocated Procedural Rules”. It was envisaged that formal descriptions associated with these rules would be used at subsequent stages of the RACM approach as a rule-based model capable of execution, to co-ordinate the run-time sequential control of EA’s in conformance with a more general structure defined by the semi-generic model.

The procedural rules play a major role during the implementation of component-based cell control systems, and particularly during the generation of application software should be developed for cell groups. The approach was later found to be capable of supporting complicated action flows in the system.
6.5.7) Services and Supportive Systems

It was realised that it would be necessary to provide a consistent information support system to underpin the modelling and run-time execution of cell controllers. Therefore, as part of the RACM approach it was decided that an information object should be attached to each system component, which defines the information requirements of that component. Information objects should be formally defined and related to each other via the “generic information model” (§6.3.2).

Furthermore it was understood that a unified interaction system is required to realise flexible and implementation independent interoperation between system entities. Such an interaction system should manage the use of communication and protocols and actual data transactions in the system. Naturally the interaction system requires system networking support. Such a network system should be able to: a) establish connectivity between system applications (represented by modelled elements); b) be easily configured to support interactions between functional and information elements (used by software components) in a given cell system; and c) distribute computer loads to available hardware resources.

The design and application of the supportive information and networking system is further discussed in appendix D.

6.6) Implementation Issues at the Semi-Generic Level

As discussed before, it was decided that implementation specifications of semi-generic model should be defined within the “detailed design” level of modelling (§5.1). It was also discussed that description of implementation specifications of cell and cell control systems cannot be completed at the semi-generic level of modelling, because modelling at this level of granularity may not be able to support detailed specifications that required for control and execution of cells. Therefore, it was decided that implementation and model execution issues should only be partially specified at the semi-generic level of modelling to provide samples and guidelines to facilitate implementation and execution of cell systems at a particular level of modelling that supports a specific industry cell. The implementation specifications should be precisely defined at the implementation description level of modelling during the modelling of cell and cell controllers at particular level.

\footnote{Care must be taken in defining the implementation level of modelling at the semi-generic level. At such degree of generality, the implementation model only provides templates and guidelines for the modelling execution. Such models should be later populated with domain-specific manufacturing data at the particular level of modelling, to be used during the modelling execution and change accommodating.}
Figure 6.9: Implementation issues at the semi-generic modelling level

As illustrated by figure 6.9, it was decided that three types of modelling elements must be formally specified to be able to cover implementation issues at the semi-generic level of modelling (i.e. within the "detailed design" level).

These modelling elements include "Reusable system components", "particular system elements" and "software component templates", as described below.

1) **Reusable system components** – In general, a "system component" is modelling representation for a block of software application that enables the control and execution of related cell activities in real world environment. Therefore "system components" are reusable modelling building blocks that describe: a) system activities at a given granularity (i.e. BP’s in this study), b) resources and information required to realise those activities, and c) infrastructure services, supporting tools and relationships amongst those modelled entities (see figure 6.9).

It was decided that these elements of system components should be defined by semi-generic model, and therefore their use could be structured and supported in a specific domain of manufacturing cells. The aim was to enable cell designers to readily reuse components when developing particular models of cells and their control systems. However, when cell requirements change, the "system components" may also require modifications or reconfigurations. In these cases "cell developers" should realise new requirements by a) reconfiguring the generic modelling entities within the "conceptual design" level of semi-generic model, and b) redefining the "system components" using the RACM approach.

It was decided that "system components" should be organised in the form of library of reusable building blocks to facilitate modelling during the system design and development. A set of
sample system components is illustrated in Appendix A.

2) **Particular system elements** – It was understood that some of the modelling elements that required to specify implementation issues cannot provide practical usability if they were defined in a generic manner. Rather these types of elements depend on the ways that a particular cell is designed, configured and implemented (i.e. abstracted by the *particular* level of modelling). It was found that within the semi-generic level of modelling, the following “particular system elements” can only be defined with references to specific cell systems:

a) modelling elements that specify “cell groups” – these elements are required when developing cell application tools (CATools - §7.5.3),

b) modelling elements that specify assignment of human resources to particular tasks – depending on the cell automation level, tasks may (or may not) be performed by human resources (e.g. manual or automated quality control task),

c) allocated procedural rules – these are required to specify the actual flow of processes for a particular “group” within a cell. The “allocate procedural rules” are required when developing cell application tools (CATools - §7.5.3)

d) modelling elements that specify “connectivity functions” – it was decided that a function code should be attached to a software component to enable interaction and connectivity with other system entities. These modelling elements may vary based on the particular infrastructure services used for a specific cell system.

e) modelling elements that specify “data manipulation functions” – it was also decided that a function code should be attached to a software component to enable manipulation of data required to accomplish tasks embedded in that software component (it was assumed that data would be located in either local or remote data storage systems).

Although it was realised that definition and configuration of “particular system elements” should be completed at *particular* level of modelling, it was found necessary to assume a set of default configurations for these elements at semi-generic level of modelling to be able to proceed the modelling constructions. This can also facilitate the development of *particular* models by providing guidelines and sample configurations. Making such assumptions were also found to be necessary during testing and evaluation of the semi-generic concepts described in this study, particularly when developing proof-of-concept application tools. Default configurations for these modelling elements are shown in Appendix A (§A.5).

3) **Software component templates** – The “reusable system components” need to be associated to “software components” which provide capabilities to perform tasks (structured in the form of

---

2 Different class of cell users will be discussed in chapter 7.
BE’s) related to each system components (see §6.5). However, the development of software components greatly depends on the ways that a specific cell and its cell controller is designed and constructed. Therefore, during semi-generic modelling, it was decided that software components would be represented by “software component templates”. The templates describe specifications required for “to-be-made” software components and provide guidelines to assist their development. The templates would be defined based on: a) description of “system components” (defined and structured by semi-generic model), and b) definition of “particular system elements”, as shown by figure 6.9.

It was decided that templates typically should define the functionality, input/output interactions and data formats deployed by software components. An example of the templates developed to support semi-generic modelling is illustrated in appendix A (§A.1).

Applying the RACM approach, it was understood that special application tools would be required to support the application of the semi-generic model during the detailed design of cell control systems. Therefore a set of prototype application tools were developed to facilitate implementation of the modelling systems. The prototype tools include a “Cell Design Tool” that facilitates the RACM approach, and the “Cell Application Tools” that provides capabilities for system end-users to control and monitor cell systems. The specification and development of prototype tools will be discussed in chapter 7.
Summary
The construction and elements of the semi-generic model of manufacturing cells and its use during conceptual design was previously discussed in chapter 5. The use of this model and its elements during the detailed design was explained in this chapter.

During detailed design, the semi-generic model helps maintain effective relationships between modelling entities and physical cells. The model was constructed to achieve this by defining functional entities. These entities represent physical elements of a cell within the developed modelling framework.

Also discussed in this chapter was the development of a model-driven component-based approach to the design and configuration of manufacturing cell controllers. This approach has been termed the “RACM approach”. A key premise on which the approach is based is that the assignment of cell resources and system components (i.e. reusable modelling building blocks of control system functionality) to appropriate grouping of cell activities. Also discussed were ways that system components should be assigned software components that provide functionality required during the run-time execution of systems. A sample set of generic modelling entities and system components that were specified and developed during this research were also discussed in this chapter.

Furthermore, the development of a semi-generic information model was considered in this chapter. It was explained that information requirements defined using the CIMOSA information modelling approach can be formally mapped onto constructs of the EXPRESS modelling language. Thereby, information objects developed as part of an EXPRESS model can be also attached to system components when deploying the RACM approach.

The remainder of this thesis describes stage two of the research (see research plan in §4.3) which were aimed at evaluating and testing the concepts developed in this study.
7.0) Introduction

In earlier chapters of this thesis a requirement for change capable manufacturing cells was considered. This led to the conceptual design and development of a set of systems modelling and integration tools and a methodology and semi-generic model to support the design, construction and change of manufacturing cells in a specific domain of manufacturing. Here a unified set of enterprise modelling, component-based design, and information modelling concepts was developed.

However it was necessary to assess the practicability of these concepts (see stage two of the research plan - §4.3). Therefore this chapter will explain how the project concepts can be applied to:

- develop models of specific cells (i.e. particular models) by populating semi-generic model with data from specific cells,
- operationalize particular models in a way that leads to the development of a change capable controller for the specific cell,
- specify the requirements of proof-of-concept tools that have a capability to facilitate use of the modelling concepts in real world environments,
- specify the requirements of a set of infrastructure services that can support the life cycle engineering of specific domain of manufacturing cells represented by the semi-generic model.

7.1) Modelling Real Cases

The semi-generic model of cells and their control systems describes common groups of activities (termed generic tasks) that need to be carried out in a specific domain of manufacturing cells. It was understood that practical constraints would exist on the use of semi-generic modelling, as it would prove difficult to specify in detail the operational tasks that should be performed during the run-time operation of cells by using the description of semi-generic tasks. For instance consider the case of a PCB cell where scheduling, dispatching, loading and device setting-up must be performed before a specific operational task such as soldering a part onto a board can be carried out. Here it was understood that neither the method by which soldering operations should be performed, nor the way that the required operation of physical devices should be integrated to the modelling descriptions, could be completely specified by a semi-generic model. A semi-generic model is concerned with the type of tasks, rather than details of them. Therefore it was concluded that the semi-generic model should encode allowable relationships between "classes" of tasks and
resources. However at some subsequent modelling stage, the real world system would need to be more completely described i.e. at a detailed modelling level or particular level of modelling as defined by CIMOSA. Therefore it was understood that particular models of cells are effective only when specifying "static systems". In the cell developed based on these models any change required in the system resources, communication methods, or other system entities will require a new cell description that should be formally developed into a particular model.

Following this line of reasoning, it was found to be practical to develop particular models in an effective and sufficiently complete way (1) by making references to generic modelling elements and relationships between such elements and (2) by particularising and configuring such a semi-generic model based on requirements of specific cells. It was understood that an important advantage gained by deriving particular models from a semi-generic model is that the operation of physical systems could be adequately supported by the modelling environment in sufficient detail, and yet could be flexibly redefined and re-configured when certain types of changes needed.

7.2) Modelling Particular Cells – The Methodology

It was assumed that a semi-generic model should define the generic system entities and their relationships for a specific domain of manufacturing cells (e.g. PCB domain). However it was understood that a modelling method should be deployed by system designers and builders that enable developing modelling of particular cells (i.e. particular models) from generic system entities.

To provide a firm foundation for modelling methods and to promote its future acceptance and use, it was decided that the structure of particular models should conform to the CIMOSA modelling approach. Furthermore, it was realised that the CIMOSA approach should be accompanied by component-based approaches to enable development of the resultant modelling system into a suitable set of inter-operating cell modules or components (see §6.4).

Figure 7.1 illustrates the structure of particular models, method of developing these models, and how it was decided that modelling at each level should be assisted by a specific set of modelling tools.

---

1 Here, a "static system" was assumed to be an operational cell system with fixed and predefined states: entities, including a set of resources (i.e. human, machine and applications), a description for retrieving required data from the data storage systems, a fixed interchanging intercommunication method, a specific set of operations defined by the process plan, and so forth.

2 It should be mentioned that some of the research groups [142] have proposed particular models with capability of allocating the system entities with optional choices. This method may improve the flexibility of the particular models to adapt with system changes in a limited domain, however based on limitations of such approaches, it was concluded that the particular models should be reconfigured (and regenerated) based on the formalisations defined by semi-generic model.
Therefore as the approach was to be based on CIMOSA modelling concepts, the modelling of particular cells would need to be carried out at three levels, namely: Requirement Definition level, Design Specification level, and Implementation level. By conforming to the CIMOSA modelling structure, consistency of the models created at these three levels could be maintained. In a manner similar to that used at the “conceptual design” level of semi-generic model, the requirement definition level of particular models was defined based on the CIMOSA modelling structure by deploying the SEWOSA method. Whereas it was understood that development of particular models at the design specification and implementation levels would require the definition and use of a new modelling method and supporting tools as none had previously been defined by the CIMOSA consortium or by subsequent users of CIMOSA modelling concepts. Therefore various design decisions had to be made and linked to prototype testing. This led on to the definition of the RACM approach (see §6.5) and hence component-based approach to engineering manufacturing cells.

When developing a particular model at the requirement definition level, it is necessary to (1) select an appropriate set of generic modelling entities as defined by the semi-generic model, (2) configure these entities into practical system structure, and (3) populate the model with the real case data. Design decisions made about the method used to develop particular models from a semi-generic model are illustrated in figure 7.1 and described in further detail below.
Initially, a set of business entities (i.e. structured format of "generic tasks") should be chosen from the list of generic entities predefined by the semi-generic model. Choice of generic tasks must maintain the operational requirements of a particular cell. The business entities should be supported with a selection of information objects required during the modelling and real-world operation of life phases. The selected sets of business entities and the information objects would also need to be populated with the real case data.

At the design specification level, it was decided that ordered subsets of business entities and their related information objects should be assigned to organised groupings of resources in a particular cell. As explained earlier, the main reason for breaking down particular cells into organised groups of resources was to handle complexity, where it was understood that each group would function both individually and collectively to realise goals of the cell. Each group of resources would need to have the necessary means (i.e. functional capabilities) required to perform activities assigned to that group. Also it was decided that the resources in each group would be modelled as "system components" (§6.6). Suitable sets of system components would also be predefined by the semi-generic model and would include definitions of machines (and their drivers), cell operators, supporting software applications, and the infrastructure services required to realise the manufacturing requirements defined by the business entities.

It was also decided that general relationships between selected modelling entities could be maintained by using a predefined structure and by using a default structural configuration also defined by the semi-generic model (see "default configuration" - §6.6). However, the pre-configured entities would still need to be populated by real case data and essentially this would result in a fine tuning (i.e. a degree of reconfiguration) of the generic default structure.

Having proposed the RACM concepts, and having specified an associated RACM method that leads to a set of inter-operating components that conform to specifications embedded into a set of CIMOSA conformant models, it was necessary to operationalize the concepts and method. Here it was understood that support for cell system designers and builders would be required in the form of software tools. Also required would be a supporting integration system capable of underpinning the runtime control and interoperation of components within a particular cell.

At the implementation modelling level, it was decided that means would need to be provided to structure and support the development of cell control applications, thereby enabling the end-user of cell control system to manage, control and monitor a particular sets of cell operations. It was assumed that this would be achieved by developing appropriate cell controller tools. Here it was decided that each tool would comprise a software kernel and a selected set of "software

---

3 For instance in the case of modelling a PCB cell, by selecting and populating a generic task "Preparing Tools and Equipment" (i.e. EA9 and EA10 from BP4/DP2 - see Appendix F) particular tasks can be defined comprising "EA9=preparing toolkit TK405" and "EA10=preparing fixture FX208" which concern a specific toolkit and piece of equipment related to the particular cell.
components” that collectively have the capability to perform the sets of activities defined by a business process (see Appendix A, §A.6). It was also decided that these software components would be developed based on the descriptions of “software component templates” specified by the semi-generic model (see §6.6, §A.6 and Table A.5). It was assumed that the software components could be programmed and generated using a variety of graphical user interface tools.

Consequently as part of this study, it was found necessary to design, deploy and develop a set of prototype tools capable of supporting the development of particular models in conformance with the RACM approach.

Therefore, a “Cell Design Tool” (CDTool) was developed to support system designers and builders of specific cell control systems. The CDTool facilitates the detailed definition and configuration of particular models. Furthermore, this tool was designed to generate a set of “Cell Application Tools” (CATools) capable of controlling operations assigned to each “cell group”. During the runtime operation of cells, the CATools were designed to interoperate with other elements of the cell including system end-users, other CATools, device drivers, and infrastructure services (see figure 7.1).

The design and development of prototype tools and the support they provide to improve “change capability” of cell systems, will be discussed later in this chapter.

7.3) Generation of Particular Models

It was discussed earlier that particular models should be built based on the descriptions defined by semi-generic model. The descriptions should be instantiated using real case data (here referred as “static data”) captured from particular industry cases. Figure 7.2 shows an example PCB assembly cell⁴. The figure also illustrates in a stepwise manner, the method used to deploy semi-generic models to develop a suitable particular model for the example cell. In the example case, the cell receives bare-boards as inputs to the cell and assembled boards as output to the next cell in an assembly line. The cell includes: a printer machine manually operated by a cell operator; automatic placement, solder and wash-off machines; and a conveyor used by an operator (referred to as inspector) who randomly (approximately 10% of boards) controls the quality of parts.

This cell can operate on different PCB products based on scheduled part lists sent to the cell by the cell supervisor. The operator should constantly monitor the status of printed boards to maintain appropriate buffer size required by the automatic devices. The inspector should also feedback the results of the quality checks to the cell supervisor to maintain the required throughput of the cell.

As discussed before and illustrated by figure 7.2, a set of generic “system entities” defined by the

---

⁴ In this research, the "static data" and "dynamic data" are assumed to be part of "engineering data" related to the system operation (and its function and information requirements). The static data include the cell specifications during the modelling life cycle, whereas dynamic data support information requirements during the system execution phase.

⁵ This example cell is made based on a case study described by Aguiar [117].
Business Entities – During the generation of particular model of cells, functional activities of that cell should be identified and classified as pre-defined semi-generic business entities. Accordingly, an appropriate set of the business entities should be selected to generate a particular model at the requirement definition level of modelling. In the example cell for instance, three domain processes (DP) would be required to meet operational requirements of the cell. However,
only some of the business processes (BP) and their subsequent enterprise activities (EA) in each DP were found to be necessary and therefore were assigned to this particular cell model.

- **Procedural Rules** – A set of “Reference Procedural Rules” have been defined and are contained in the semi-generic model (§6.5.6). These rules specify process flows related to generic enterprise activities within a specific domain process. The rules are based on CIMOSA Event and Status entities and are defined as the input and output messages sent/received by enterprise activities. When developing the particular model for the example cell, specific selections of these rules should be generated in accordance with the selected set of business entities for the model. Therefore the process flow between enterprise activities can be redefined and “Allocated Procedural Rules” should be produced. The “allocated procedural rules” should be attached to particular cell models at the requirement definition level of modelling.

- **Functional Entities** – Generic types of functional entities are also defined by semi-generic models. It was decided that functional entities of a particular cell should be defined and populated for three categories of resource, namely: human resources, machine resources, and application tools. The definition and configuration of these entities should be carried out as part of particular model of cells at the design specification level of modelling. For instance, the printer machine is a physical device function entity and its operator is human resource function entity as defined for the example cell.

- **Particular System Elements** – These elements can only be defined based on the particular cell specifications and requirement (see §6.6), such as dividing cell into “groups”. When defining cell groups, a selected set of functional entities, business entities, and “allocated procedural rules” should be assigned to the groups. For instance, the example cell includes 3 groups as illustrated in the figure. The “supervisor group” should be assigned all BP’s within the DP1 and monitoring BP from DP3. Furthermore, appropriate sets of “system components” (see §6.6) should be assigned to the groups, based on the allocated BP’s to each group.

- **Information Objects** – It was decided that general information requirements of cells are specified by the “conceptual design” part of the semi-generic model. This is subsequently used to structure the design of information models at the “detailed design” level of modelling. Specific instances of the information model should be populated by specific case data (§6.3.2) at the design specification level modelling. The “system components” defined for each BP includes definition of information objects required for that BP.

---

6 According to the definition given by CIMOSA, a functional operation (or in general enterprise activity) can be triggered by various Events and therefore its initial Status will change to its final Status. A sample set of Events and Status items have been used in this study as illustrated in appendix A (see §A.4).
• **Software Components** – When developing the implementation part of the semi-generic model, a set of “software components” were pre-defined to provide functionality required for the “system components” during the run-time operation phase of the cell system. Subsequent to the assignment of the system components, the software components should be allocated to each cell group. For instance in the “group Operator” in the example cell, software components SC201 to SC206 (see table A.5) are assumed to operationalize the BP’s 201 to 206 (i.e. receiving new batch, data preparation, printer set-up, etc.).

• **Infrastructure Services** – Each system component has been designed to include embedded codes that enables it to use the integration and information services. These services should be reconfigured based on the software component allocated to each cell group.

It was found necessary that software components and their supporting infrastructure services be associated with a kernel software programme that co-ordinates the use of these elements when generating the actual cell control application tools (CATools) for each group. During the run-time operation of the cell, the CATools should be connected to the physical system resources (or the resource drivers).

### 7.4) Development of Prototype Application Tools

A general characteristic of modelling methods is that they use modelling constructs to represent real world entities in a descriptive format [158]. Typically computer tools are required to facilitate understanding and processing of these constructs in a logical manner⁷.

In addition, computer modelling tools can enable human modellers and analysts to assess the effectiveness of the models (e.g. their completeness in representing the real world) and validate the accuracy of the models and the systems represented by those models. When developing modelling support for an existing manufacturing enterprise, modelling tools should facilitate the use of models, and also establish links between the modelling environment and the real-world systems, including existing resources and application tools.

In this research, a prototype application toolset was developed to facilitate assessment of the practicality of modelling concepts developed by this study. The aim here was to carry out both pilot studies (based on hypothetical cell systems created in the laboratory) and industrial case studies models (based on real industrial cases).

The prototype application toolset was designed and developed for three classes of users that are typically involved in the design and construction of manufacturing cell systems. The user classes

---

⁷ Many industrial examples show that the descriptive models have only been used as guidelines or reference to help analyse the performance of real world systems. However, potentially they can also provide constant support during the design and operation of these systems, providing they are accompanied with appropriate computer applications that integrate the modelling and real world environments.
Figure 7.3: Overview of the “Prototype Toolset” developed in this study include manufacturing System Developers, System Designers, and System End-Users.

Figure 7.3 illustrates the prototype application toolset developed in this study. The toolset comprises the following types of application tool:

- **Semi-Generic Modelling Tools** – These tools will be used by manufacturing “system developers” to design and construct semi-generic models.

- **Cell Design Tool** – “Cell system designers” will generate (or re-configure) particular models by using the “Cell Design Tool (CDTool)”. This tool enables its users to: a) capture real case information and instantiate the semi-generic model; b) apply the structural guidelines defined by semi-generic model when developing particular models; and c) produce applications used to execute particular models.

- **Cell Application Tools** – The “cell system end-users” will utilise “Cell Application Tools (CATools)” to operationalize modelling systems during its run-time operation.

The specifications of the prototype tools are discussed in the following sub-sections. In addition, appendix D provides details related to the development of these tools.

### 7.4.1) Semi-Generic Modelling Tools

As explained in the foregoing and illustrated by Figure 7.1 a set of system modelling and construction tools was required to develop and operationalize the semi-generic model of PCB manufacturing cells, and thereby to facilitate testing of concepts proposed by this research. Also as explained earlier essentially the semi-generic model was designed to have two parts, one concerned primarily with cell design issues, the other primarily with implementation issues. During the study it became evident that there were a number of existing modelling methods and supporting tools that could be used to support much of the development, operation and testing of the first part of the semi-generic model. But with respect to the development, operation and testing of the second part new methods and tools had to be devised and developed, largely because the implementation part
of the semi-generic model was based on component-based engineering concepts developed within this study.

As part of the semi-generic modelling tools, the SEWOSA [117] and CIMBIOSYS\(^9\) [139] applications have been adopted and used in this research to develop the functional views of the semi-generic model at the conceptual level. The SEWOSA CASE tool models system specifications in the form of structured text format, and graphically represents different aspects of the models in the form of various types of diagrams. Appendixes E and F discuss the use of SEWOSA tool and illustrate the semi-generic model developed in this research using this CASE tool.

In addition as discussed in section 6.3.2, the EXPRESS information modelling methods (and tools) have been used as part of the “semi-generic modelling tools” to develop the information aspects of the modelling systems. The semi-generic information model developed in this research can be populated by the real-case data to structure the information system required during the modelling and real-world life cycle of cell systems. Here, the development of information system was facilitated by the use of an EXPRESS/STEP parser (see §7.5.2), a database management system (such as INGRES database management system used in this study) [99, 141, 154, 159] and an interaction and data exchange facilitator (i.e. CIMBIOSYS).

Furthermore, the “semi-generic modelling tools” also include a graphical user interface (GUI) builder. The specifications of a set of generic software components have been defined by the implementation part of the semi-generic models (see §A.6). A set of sample software component has been defined to enable proceeding the development of prototype tools. Many commercially available (GUI) builders can be used to generate software components. The selected tool for this purpose should be able to develop object-oriented applications with capability to construct user interface in a modular structure. Here a number of GUI builders have been considered (see Table A.5). In this research the “SUIT” [160] user interface builder was chosen and implemented as explained in Appendix D. The information related to the structure of system building blocks, pre-defined software components and associated information objects should be accommodated in a data storage system as part of the semi-generic model (as indicated by “DB 1” in figure 7.3).

\(^9\) At the semi-generic level of modelling, the CIMBIOSYS tool was only used to provide sufficient interactions with data storage system.
The “Cell Design Tool” (CDTool) is the main prototype application tool, developed in this research. This tool is used to generate, reconfigure and operationalize particular models, when certain type of change occurs. This is achieved by supporting an application of the component-based engineering concepts developed during this study. The types of change that can be supported by the CDTool will be discussed later.

The CDTool has been designed in three modules. In the first module, the tool retrieves information about the semi-generic system entities, maintained by the library of the reusable building blocks within the semi-generic model. This information, including descriptions of the FE’s, BE’s and associated information objects and the procedural rules, should be used as guidelines or “templates” to generate particular models. The requirement definition level of the particular models will be structured based on this information (as shown by figure 7.4 - module 1).

The second module includes capabilities to support the cell designer to define cell “groups” and assign cell resources (§5.2). The tool enables a system designer to select required business entities and assign them to designated cell groups, and accordingly generate “allocated procedural rules”. At the same time, the tool populates the functional entities with the particular modelling case data, and allocates them to the cell “groups”. The allocation of modelling entities, and the provision of
information/networking support corresponds to the design specification level of modelling a particular cell (see figure 7.4 - module 2).

It was found that the utility of this second module would be heavily dependent on the knowledge and expertise of the system designer who would need to have an intimate knowledge of cell requirements in terms of necessary support required by both the modelling entities and the cell physical resources needed to realise functional tasks.

Finally, the third module of the CDTool was designed to generate the “Cell Application Tools” for the system end-users. Completing the configuration of the modelling entities and access to external supporting tools (e.g. scheduler and database software), the CDTool specifies the information objects required for operating the selected business entities and retrieves them from the semi-generic information model. Furthermore, the CDTool specifies the required set of software components associated to the business entities selected and allocated to each “group” in a particular cell.

In the module 3, the CDTool generates one “cell application tool” for each “group” in the cell, based on the a) selected software components, b) a predefined software kernel to co-ordinate the software components and provide the overall user interfaces, c) associated network configurations, database accesses and user preferences.

The predefined software kernel is common for all CATools and was designed to provide an interaction system that underpins inter-working between software components and external systems elements (e.g. other information and application systems in the host environment of the cell).

The user preferences have been implemented in this research with respect to tasks that are specified for different classes of users, e.g. cell supervisor or cell operators. However, such preferences might be developed up to provide custom built user interfaces. Such an opportunity has been a subject of investigation in another research project [23, 24] in MSI Research Institute.

The user preference issue will be further discussed in §9.4.

Essentially module 3 of CDTool develops the particular model at a level that corresponds to the implementation level of CIMOSA modelling.

7.4.3) Cell Application Tools
In this research common units of functional activity in a domain of cell systems (in this case PCB manufacturing domain) were defined as semi-generic tasks that can be structured in an organised form into business entities. The information, function, and human resources required for
performing a particular set of tasks (i.e. business processes) were described as "system components". Subsequently, the "software components" were defined to operationalize the "system components" in a physical cell environment. The cell systems were broken down into "groups". Each group would be assigned responsibility for meeting a subset of requirements (namely some selected portion of a BE) and would be allocated a suitable set of resources (i.e. components, including software components) to achieve that subset of requirements. In this study it was decided that the CDTool would support the development of a software capability (i.e. CATool) for each cell group. This was achieved by wrapping up the software components (and their functional and information capabilities) along with the infrastructure capabilities required to perform the tasks (i.e. portion of a BE).

Therefore it is the purpose of CATools to carry out cell control and monitoring processes during the runtime operation of a cell and provide operational support for "system end-users". These end-users are in fact the human resources allocated to each cell "group" at the design specification level of the particular models (i.e. module 2 of the CDTool).

Therefore the CATools provide a human interface to the component-based cell control system. This is achieved by defining and establishing interaction links with physical device controllers, other cell entities and their information systems, external entities (such as a scheduler), and also other CATools.

Depending on the level of automation specified for a particular cell, CATools will need to co-ordinate, control and monitor the performance of cell tasks. This requires the CATools to provide appropriate device drivers, receiving new order lists, schedule and distribute tasks between groups, dealing with unexpected events, generate reports, and so on.

When developing a particular cell model, at the design specification level the CDTool was designed to generate two types of CATool. The first type has been designed to support the cell supervisor, by enabling him/her to co-ordinate and distribute tasks to cell groups which should be carried out within the scope of the cell, and to reschedule the cell and dispatch new jobs, as required. The second type of CATool has been designed to support operators and enables them to receive, control and monitor the jobs carried out by cell groups for which they have a designated responsibility.

As discussed earlier, a variety of commercial application tools could have been modified and used to generate software component building blocks of the RACM approach. However the compatibility of available tools with respect to RACM component concepts needs to be assessed to determine the extent of that modification and the likely constraints that their use might place on the

---

11 The level of operator control over the system depends on the degree of automation designed for the cell system. The author has considered an "operator confirmation" activity before starting the "run a job" enterprise activity, which practically requires physical execution to be initiated consequent on confirmation by a human operator.
approach. Here, appropriate software architecture (as should be defined by software vendors) can be deployed to maintain the reconfigurability provided by the modelling structure when generating the CATools. Therefore as part of the prototype CATools development and testing in this research a selection of sample CATools were created using the SUIT user interface tool. This work is discussed in appendix D (§D.6).

7.5) Infrastructure Services for the Prototype Tools
To be able to test the practicability of the concepts developed by this research using the prototype toolset, a number of infrastructure services were required. In this research the required services included the interaction systems between the cell elements, and the information systems developed for the modelling and real world system operations. The development of these services is discussed in the following sections.

7.5.1) Supportive Integration System
Much has been said about the importance of integration systems in the manufacturing environments. Consensus views of experts on integration issues in manufacturing systems have been outlined in the literature review (§2.5). In general, integration systems are required in this research to enable cell system entities (modelling and real-world entities) to interact with each other and exchange system data. In this context the important issues considered when adopting an integration system for this research were its ability to:

a) provide standard techniques to format the data that should be exchanged,

b) add knowledge about ways of accessing data whilst concealing technical details,

c) exchange data and control messages between two system elements, using a unified language or deploying specially designed application drivers or interfaces.

Arguably, many of the available integration methods including CORBA [58], Internet-based interactions [109, 161], KQML [112], and CIMBIOSYS [139], provide capabilities required for this research. However, to be consistent with the other applications developed in MSI Research Institute and used in this study, the author has implemented the CIMBIOSYS integration method to establish the integration system within the modelling environment. CIMBIOSYS provides underlying interaction methods required to underpin communication amongst cell system entities. During the development of software components, appropriate CIMBIOSYS functions need to be attached to the program codes. This enables the cell integration system to communicate with the software components and send/receive data requests, embedded into command messages.

For existing application tools within the modelling system (e.g. scheduler, database), appropriate

\[12\] The author has experienced the alternative methods such as internet-based integration systems in the other research projects [161] and believes that such modern methods can significantly improve the capability of the MCC models. However, considering that the integration system is not the main issue of this study, and the use of CIMBIOSYS meets the requirements of the prototype applications in this research, therefore the integration system has not been changed or upgraded with the modern methods.
CIMBIOSYS drivers were developed to enable consistence interaction with these application tools. The technical implementation of the CIMBIOSYS method has been discussed in Appendix D §D.7).

7.5.2) Supportive Information System

In conformance with the information view of CIMOSA at the semi-generic level of modelling, information requirements of cells in the PCB domain were defined in the form of reusable information entities (§5.3.2.2). Accordingly an information model was designed and coded using the EXPRESS language. The conceptual schema for the information model was defined in a way that allows internal schema to be accessed using the SQL language. This enables the information system to manipulate physical system data. Finally, the information objects (i.e. EXPRESS information entities) were classified as part of the reusable building blocks.

The information objects so defined specify generic information classes required for a domain of manufacturing cell systems. However, in a particular level of modelling, this information needs to be fully specified by:

a) populating particular information objects with the real-case data (here called “core-data”),
b) wrapping the “core-data” with the appropriate information, required to access core-data stored within the data storage system (here called “accessing information”), and
c) appending information about the method of transferring the “core-data” from one place to another.

At the implementation level of modelling, the information model should specify the “information system” required to support development and use of a particular model. Here the information system is the actual methods and tools needed to provide the data required by physical cell resources and modelling entities in a particular cell.

It must be emphasised however that as a result of using the information modelling approach developed, the functionality attributes associated with information entities will not form as part of resultant EXPRESS models. It follows that during design specification modelling the semi generic information model encodes only a single view of (or perspective on) manufacturing cells, i.e., it only describes information exchanged among cell elements. It ignores other issues, such as where a block of information comes from and who or what element will make use of it. To illustrate this point, consider the case of a scheduler package that receives new order information, generates new work plans for a cell and sends them to a cell supervisor. The issue of concern here is solely about the information block exchanged between the scheduler and supervisor. Each information block is a specific class of information object, which can be used (or generated) by different authorised elements of a cell.

Figure 7.5 illustrates the information system developed in this study to support information
requirements during the modelling and real-time operation of particular cell systems.

Initially the EXPRESS models developed as part of this study were used to construct the structure of a more specific data storage system within the particular information system. In addition a set of specific "real-case" data was populated into the system. Here it was understood that the information could be stored in individual local files or in a central database system.

It was found that by implementing a central database system better consistency could be achieved when exchanging data between databases and particular cell elements.

The SQL language was selected as a standard interface to database systems (see literature review §2.7). Here physical data was formatted within a STEP physical file [100] whilst data entities were structured according to the EXPRESS model.

There was a need for a SQL translator application (i.e. an SQL driver). This application was required to enable communication with system databases by generating a set of SQL statements and building appropriate database tables. The tables were also populated with data formatted as STEP physical files.

Such an SQL translator had previously been developed by other members in the MSI Research Institute [141], hence this tool was used in this study. However, currently several EXPRESS translator tools [99] are available in commercial and educational bases, which could also have been used to facilitate the implementation and modification of the developed information system.

Figure 7.5 illustrates the mechanism used to facilitate interaction between an example CATool and a central data storage system. To provide capabilities for data exchange using an integration system, SQL driver codes and CIMBIOSYS functions were embedded into the CATools (see
Appendix D for more details). The figure depicts an example of retrieving the *object view OV-11* (new order list) of the *enterprise object EO-1* (ordering management - see Appendix C) from the database through the database driver and the integration system.

In practice, appropriate *function operations* (FO’s) within the software components (embedded in the CATools) transmit a request message to the database driver service. The FO’s send a CIMBIOSYS message with a specific information object identification (i.e. “accessing information) and receive the value of the requested information (i.e. “core-data”) following transactions of several CIMBIOSYS functions (one function for each record of data contained in a table).

### 7.6) Modelling Support for Operational Processes

It was understood that modelling approaches developed and used in this research should be able to improve the “change capability” of manufacturing cells by facilitating the reconfiguration of system elements and their relationships. Different types of change in manufacturing systems were considered and classified by the “Model of Change” discussed in chapter one (§1.2.1 and figure 1.2). These include (1) changes in business requirements, (2) change in production requirements, and (3) change in operational details.

It was envisaged that changes of type (1) will typically be of concern when defining manufacturing systems requirements and goals. However it was not prime concern of this study. Whereas changes of type (2) classifies changes in manufacturing systems that potentially can be supported by the modelling system developed in this research. In addition, it was also envisaged that changes of type (3) does not typically occur in the domain of manufacturing systems that has been focused by this study. Change capability and the ability of the modelling system developed in this study to support various types of change will be comprehensively discussed in chapter 9.

However, it was realised that changes of type (3), referred as change in “operational details”, occurs on a constant basis during the run-time operation of cell systems. Therefore, it would be necessary for the modelling system to be capable of linking to the other manufacturing support systems that provide support for change of type (3).
This section describes assumptions made about how the modelling structure (and its support tools) should be linked to the real time operation of cell entities and thereby the nature of the support that should be provided by particular models in order to specify operational change.

The following assumptions were made about the run-time execution of a cell system. Tasks in a cell are carried out by sequentially performing a number of "job" sets. A job set comprises a number of jobs that require a similar set of system resources and information in order to be carried out. A "job" is the actual activity that should be performed on a product or a service (or an information object) to transform it from one status to another. Examples of job classes are machining, painting, washing, etc. A "job", in turn, comprises a number of "operations". Here an operation\(^{13}\) can be viewed as the smallest functional unit of activity that should be processed when accomplishing a "job". To process a job all related "engineering data" should be available. The "engineering data"\(^ {14}\) is a set of information required by a human operator or automated devices to perform an "operation". This data should typically be provided by a process plan. The engineering data should also specify the flow of operations within a "job". This set of data can specify various flows of operations, based on the "termination status" of an operation.

\(^{13}\) For instance a "job" could be producing a shaft and every step of the related process plan will be an "operation".

\(^{14}\)
Figure 7.7 illustrates part of a particular model developed for an example cell that includes two “groups”. Group 1 (i.e. supervisor group) has been allocated a set of tasks that are specified by business processes 1, 2, and 4 from domain process 1 (see the figure and also appendix F). These tasks include: “receiving, scheduling, and dispatching” new orders to the cell. This “group” of cell produces the “job lists” to be operated by resources classified in the other cell groups. For instance, when a specific set of jobs was dispatched to “group 2”, the CATool that operates and controls tasks in this group will receive the “job list”, retrieve the required engineering data, and specify a series of jobs (a job may be operated once or repetitively, depending on the part batch size). The tasks defined by BP3 (from DP2) should be carried out by group 2.

Subsequently, the job series defined within the BP3 are sent to BP4 (from DP2), where the machine, tools and other operational requirements for a particular job are being prepared. Following BP4, BP5 should be executed to actually operate a “job” and monitor the status of the system elements (and parts) after this operation. The process of retrieving, preparing and operating jobs (i.e. BP3 to BP5) should be repeatedly continued to complete the “job list”.

In this study, BP5 (in DP2) is in fact the main link between the modelling environment and the actual system resource controllers. Here the enterprise activity “Run a Job” (EA14 from BP5 in DP2) has been referred to as a “key activity”. It was assumed that when enterprise activities in a particular model are executed, all information required to perform a specific set of “operations” within a “job” should be available and a job can be operated, providing the required interfaces. Despite the fact that integration between the modelling system and actual cell physical devices is between modelling system and physical resources (e.g. the actual cell devices and machines) are prepared.

Despite the fact that integration between the modelling system and actual cell physical devices is not included in the scope of this research, it was envisaged that a variety of available methods and techniques developed by other research groups could support the engineering of such links between the modelling environment and the cell device controllers.

In general, it was found that change of type (3) typically arises as a result of activities performed by external system elements (such as a scheduler and process planner) and can be represented as a variation in a “job list”, engineering data and interaction links (e.g. database and network accesses information), etc. Here, based on the method explained above, it was concluded that changes of type 3 affect system elements at a level that typically will have insignificant influence on the execution of a particular model (i.e. the changes occur at a more detailed level than defined by a

---

14 The author has put less emphasis on the manipulation of physical resource during cell operation. However, there are other research groups (such as the research conducted by the University of Navarra [162]) that have carried out studies on this phase of the manufacturing life cycle. Collaborative research between these groups and this research can effectively extend the modelling support to the run-time execution phase. Should this collaboration research carry out, “key activity” (i.e. EA14) will be a connection point to integrate this work to those research studies. For more information see section “Future Works” at the end of this thesis.
particular model).

Summary
This chapter discussed the development and use of modelling support systems developed during this research with respect to their application in real-case manufacturing cells.

It was explained that a semi-generic model can be used to provide high level system configurability if it is deployed in conjunction with a modelling structure and reusable system building blocks. Whereas so called (by CIMOSA) particular models provide more detailed modelling support for specific “real-case” manufacturing cells. Therefore semi-generic models should be instantiated by real-case data to develop particular models.

The chapter also explained how particular models should be constructed at three levels, namely: requirement definition level; design specification level; and implementation level. Here it was explained that the existing modelling approaches could be used at the requirement definition level, whereas a new RACM approach was required and implemented to accomplish the model construction during design specification and implementation levels by developing and using a component-based engineering method.

Also explained in this chapter was the development of a set of proof-of-concept application tools to facilitate test of practicability of the modelling concepts. This prototype toolset can support construction of semi-generic and particular models and enable change capability of the overall system by generating a set of model-driven component-based cell control application tools to operate and monitor specific manufacturing cells. The prototype toolset includes a number of standard tools, applications developed by other research groups, and tools developed in this research. The applications developed in this research include: Cell Application Tools (CATools), used to operate and control cell “groups”; and Cell Design Tool (CDTool), used to construct the particular models and generate CATools.

Various types of change in manufacturing cells were also briefly discussed in this chapter and the assumptions made on the way that particular models should be linked to the other manufacturing support systems to accommodate operational changes was explained.

The application of particular models will be demonstrated in a greater detail, by investigating a case study in the next chapter. The development of the prototype tools described in this chapter is considered in detail in appendix D.
Chapter 8

Case Study

8.0) Introduction
The design and development of a modelling environment for manufacturing cell controllers was considered earlier in this thesis. Also discussed were ways in which a set of prototype tools have been developed for use in conjunction with the modelling environment to semi-automatically generate control system components which organise, co-ordinate and monitor the interoperation of real resources in a cell. Hence the tools were designed and developed as a means of enacting enterprise modelling concepts. It follows that collectively the modelling environment and support tools provide an environment that can be used to evaluate benefits gained from operationalizing enterprise modelling concepts. This chapter reports on such an evaluation exercise where the toolset was used to enact enterprise models, thereby supporting the unified design and implementation of a real world case study system.

Also described earlier in this thesis was the way in which this research was initially influenced by business and operational requirements of PCB manufacturing cells used by D2D. However, having developed the modelling approach with D2D cell control requirements in mind it proved difficult to gain sufficient access to company data to conduct a full case study based on their current and possible future practices. Also the actual configurations of cell resources currently deployed in D2D cells was found not to be ideal in terms of fully testing the use of the component-based systems engineering approach developed in this research. Therefore a decision was taken to evaluate the use of the project concepts and tools in an alternative and more accessible manufacturing domain. This led to a case study based on manufacturing cells used to produce in small and medium sized batches, high-pressure hydraulic components. Although this company is located in Iran the author had previously been responsible for the design and development of cells within that company and has direct access to company information as required. Choice of this experimental study was seen to have two advantages with respect to corresponding D2D studies. Firstly, the case study could more thoroughly test the capabilities embedded into the modelling system under close to real plant conditions and sufficient data was readily available to enable comprehensive tests to be performed. Secondly, use of a semi-generic model of PCB cells could be tested in a distinctly different type of manufacturing domain to that for which it was originally designed. This enabled (at least in part) testing of the “generality” of the modelling and model enactment approach.
8.1) Description of the Case Study Cell

The SHB Company\(^1\) is located near Tehran (Iran) and is a manufacturer of high-pressure vessels, pipes, couplings and various types of hydraulic equipment. The company has approximately 800 employees. Each year it produces circa 60 large high pressure vessels, 500 kilometres of gas pipes in different sizes, and a large number of different types of hydraulic equipment (including 2000 hydraulic cylinders). The company sells 60% of its products to companies operating in the oil industry in Iran, and 20% of its products are exported to Europe (especially its hydraulic components).

The company has several production lines, which deploy metal forming, machining, forging processes and associate materials handling processes. Within the manufacturing section, a cellular manufacturing system has been implemented to cope with a broad range of product types and production order quantities (mainly in small quantities). The cells have been organised in a process-oriented way. The manufacturing section deploys several cells, mainly cutting, machining, heat-treatment and finishing cells. The case study is based on manufacturing cells used during the manufacture of high-pressure hydraulic cylinders. Although the processes involved in this part of the manufacturing section are typically not particularly complex, the variety of product specifications (size, tolerances, surface quality), and the high rate at which orders impact on production, makes this section one of the most problematic manufacturing lines used by the company. These problems are a source of many production delays and conflicts.

A sample product and typical processes carried out by the “Hydraulic Cylinder” production line are illustrated by figure 8.1. Raw material is delivered to the “cutting cell” (typically in the form of steel bars and thick cylinders) where the material is cut and roughly sized. The cutting cell includes three conventional and two precision cutting machines. The cell receives material inputs from the material store and receives daily orders (i.e. information inputs) from the shop-floor manager. Sized material is passed to the “heat treatment cell” which carries out “stress relief” processes. Following which the material is retained in an intermediate buffer that is located between heat treatment and machining cells.

The “cylindrical machining” cell (referred to in the following text for simplicity as the “machining cell”) carries out the main machining processes required during the production of hydraulic cylinders. Therefore the design of the case study cell within system was centred on this machining cell.

---

\(^1\) The author worked with the SHB Company as a shop-floor engineer, during the period 1988 to 1991 and has detailed technical information about the production lines in this company. However, technical information given in this test case was refreshed during October 1996, when the author visited the company. Therefore this information is valid based on company conditions at that time.
The cell includes: four copy-lathe machines capable of machining parts between 30 to 350 mm in diameter, and 100 to 2200 mm in length; three CNC lathe machines capable of machining parts up to 260 and 1850 mm in diameter and length which has a tool magazine with 32 tools including a side drilling head; two horizontal boring machines capable of internally boring cylinders with a maximum diameter of 320 mm and length of 2000 mm. Material is manually loaded into copy-lathe machines of the cell so that they perform “rough and finish turning” processes on external surfaces of cylinders (see figure 8.1). Subsequently parts are loaded into the CNC lathe machines that perform “face turning”, “threading” and “chamfering” processes on both ends of the cylinder. Part unloading from CNC machines, part loading into boring machines and “boring” processes are then carried out to complete the sequence of machining operations.

Having completed machining processes, parts are transferred to the “surface finishing cell”. This “finishing cell” carries out a “surface hardening (coating)” process, that improves the surface quality and protects against abrasion and erosion. Surface hardening is followed by a “honing” process that is carried out on the internal surface of the cylinders and is performed by two vertical hydraulic-stroke honing machines. Alternatively, for parts that are less than 300 mm in length, a “grinding” process is typically involved. Finally, a sample of parts is transported to the “quality control cell” for checking.

In this company some of the resource elements of the manufacturing cells are connected to a computer network system (a Local Area Network with connections established by Windows NT
platforms). The network and associated network management system enable and control access to a central database system that maintains and manipulates orders and production data, thereby providing authorised access to manufacturing information. A network scheduler (i.e. a software tool) is also used to support the manual definition of input and output files in specific formats.

8.2) Use of the Modelling and Model Enactment Environment

The modelling and model enactment environment (which implements the concepts studied and developed in this research) was configured to support the design and implementation of the case study cell. The modelling structure deployed had three levels, based on the particular modelling structure defined in section 7.2. The following description “walks through” the basic procedure used to design and configure the modelling support system for the “Hydraulic Cylinder Machining” cell.

1) At the requirement definition level of modelling, the project methodology conforms to the CIMOSA approach. A set of business entities (representing the cell tasks) was defined with respect to the tasks that need to be carried out by the case study cell. A particular selection of

Context Diagram: Hydraulic Cylinder Machining Cell

Structure Diagram: New Tasks management DP1

Structure Diagram: Operate Tasks DP2

Figure 8.2: The requirement definition level of the particular model, designed for the test case.
business entities (BE) was made with reference to the “generic entities” defined by the semi­
generic model. The business entities defined in this way are illustrated by figure 8.2 and are
presented in the form of SEWOSA diagrams. The SEWOSA tool also generates a textual form
of this model, (i.e. model templates – see Appendix B) which was used to transfer the model
descriptions to the next level of modelling.

2) Procedural rules also need to be particularised based on an understanding of BE relationships
within the actual machining cell (§7.3). Therefore “allocated procedural rules” were defined
and assigned to selected enterprise activities for this cell. This design task was carried out with
support from the “semi­generic modelling Tools” (§7.4.1), which was developed as part of the
prototype toolset. However, the “allocated procedural rules” need to be further particularised
to suit individual requirements of cell groups. This was achieved at the design specification
level with support from the “CDTool” (§7.4.2).

3) In addition to defining functional entities, information entities must also be defined, populated
with the real data, and organised as part of the design of an overall data storage system.
Therefore the generic information elements (IE), information object views (OV), and enterprise
objects (EO) defined earlier as part of the semi­generic model (§5.3.2.2 and Appendix A) were
partially populated with data related to the actual machining cell. Initially only those
information entities were considered, which were required for the selected business entities at
the step 1. Based on the generic information entities defined in Appendix C, the information
entities chosen included enterprise objects EO-01, EO-02 and EO-03. These EO’s were
subsequently detailed to include object views OV-11 to OV-36 and their related information
elements. Here the OV’s are the information objects that should be attached to atomic
functional objects (i.e. enterprise activities).

4) At the design specification level, the particular model was developed using a component-based
approach (see RACM approach – §6.5). Initially at this level of modelling suitable modelling
entities representing cell resources should be selected (from the library of available resource
types in semi­generic model) and populated with real data (§6.5.1). Table 8.1 classifies the
resources selected and assigned with respect to the case study “machining cell”.

5) At this stage of the design specification generic function entities were populated with reference
to capability requirements of the business entities assigned to this cell. Table 8.2 shows the
function entities specified in this case.

6) As explained in section 6.5.2, the project methodology supports a decomposition of cells into
“groups” that organise and facilitate control of the resources in the cell and thereby cell
operations by applying appropriate “CATools”. These applications were generated
individually for each “cell group”.

144
<table>
<thead>
<tr>
<th>Cell Resources</th>
<th>Resource Name</th>
<th>Resource ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Resource</td>
<td>I supervisor 3 lathe-copy operators 2 lathe-CNC operators 2 boring operators 1 assistant operator</td>
<td>HSI HCO1 &amp; HCO2 &amp; HCO3 HNO1 &amp; HNO2 HBO1 &amp; HBO2 HAO1</td>
</tr>
<tr>
<td>Machine Resource</td>
<td>4 copy-turning machines 2 CNC turning machines 2 horizontal boring machines</td>
<td>MCT1 to MCT4 MNT1 &amp; MNT2 MB1 &amp; MB2</td>
</tr>
</tbody>
</table>

Table 8.1: Resource classification for the "hydraulic cylinder machining" cell

<table>
<thead>
<tr>
<th>Generic Function Entities</th>
<th>Particular Function Entities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordering System</td>
<td>Manual</td>
<td>Performs manually by HSI</td>
</tr>
<tr>
<td>Cell Supervisor</td>
<td>HSI</td>
<td>Cell supervisor receives the job-list (W-T-L) and distributes within the cell</td>
</tr>
<tr>
<td>Cell Operator</td>
<td>HNO1 &amp; HNO2 HBO1 &amp; HBO2</td>
<td>This function entity was instantiated by 4 cell operators who use this modelling entity when using the modelling system.</td>
</tr>
<tr>
<td>Scheduler/Dispatcher</td>
<td>Scheduler Driver</td>
<td>This is a driver (interface) to the scheduler software available to the cell supervisor (This software was also used as the master scheduler of the company).</td>
</tr>
<tr>
<td>Supervisor Application</td>
<td>CATool-1</td>
<td>Cell application tool was developed for the cell supervisor.</td>
</tr>
<tr>
<td>Operator Application</td>
<td>CATool-2 (A&amp;B) CATool-3 (A&amp;B)</td>
<td>These cell application tools were developed for each type of operators (CATool-1 and 2) and were individualised for each user based on the user's properties and access restrictions (version A &amp; B).</td>
</tr>
<tr>
<td>Physical Devices</td>
<td>Not implemented in this cell.</td>
<td>In this test case, the modelling system was not physically linked to the execution system.</td>
</tr>
</tbody>
</table>

Table 8.2: Populating the generic function entities with the test case data

<table>
<thead>
<tr>
<th>Cell Groups</th>
<th>Domain Process</th>
<th>Business Process</th>
<th>Enterprise Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1 (Supervisor gr.)</td>
<td>DP-1</td>
<td>BP-1 to BP-6</td>
<td>EA1 to EA11 &amp; EA14</td>
</tr>
<tr>
<td>Group 2 (Operator gr.)</td>
<td>DP-2</td>
<td>BP-1 to BP-5</td>
<td>EA2 to EA17</td>
</tr>
<tr>
<td>Group 3 (Operator gr.)</td>
<td>DP-2</td>
<td>BP-1 to BP-5</td>
<td>EA2 to EA17</td>
</tr>
</tbody>
</table>

Table 8.3: Allocation of the business entities to the Machining Cell Groups
Figure 8.3: Allocation of the modelling entities to the groups of the Machining Cell.

Modelling entities particularised for the “machining cell” were allocated to each group in the manner illustrated by Figure 6.7. For the case study the “machining cell” was divided into three groups, including a “supervisor group” and two “operator groups”. Choice here was based on an appraisal of the physical cell layout and an understanding of the similarity of processes concerned. Each group was allocated: a) a related set of business entities, which are the actual tasks which should be carried out by each group in the cell; b) a set of cell resources capable of performing the tasks; and c) a suitable set of CATools with a capability to control and monitor system elements within the groups. The business entities selected during step 1 (see figure 8.2) were allocated to each group in the “machining cell” (as shown in Table 8.3). The tasks involved in machining the hydraulic cylinders are specified by enterprise activity definitions listed in this table. In addition, descriptions of the business entities (also shown in the table) and their requirements were used to select corresponding “software components” previously specified at the “detailed design level”. Model definitions in respect to each group in the case study cell are illustrated by figure 8.3.

7) Having formally defined the functional elements of the cell system, its information aspects should also be modelled. For the case study system information objects were specified during step 3 when model construction was carried out at a requirement definition level of modelling. However, at the design specification level, a particular EXPRESS information model with a structure previously defined (see §6.3.2 and Appendix C), was generated and populated using case study data. Figure 8.4 shows part of the EXPRESS information model developed in this way by making reference to the generic information model and “machining cell” data. The information model generation was much simplified by having knowledge of selected object views required for the cell. The object views were themselves chosen with reference to the business entities defined during step 1. As mentioned earlier (§6.3.1), CIMOSA-based information entities were converted into equivalent EXPRESS modelling elements.
During this step, construction of the particular model was continued at the implementation level of modelling. Here CATools (§7.4.3) were generated for each group within the cell, with reference to the requirements of business entities allocated to each group. Each CATool designed for a group comprised a number of “software components” which conform to the previously specified “system component” (see §6.6) defined by the semi-generic model. Therefore the software components were selected based on capability requirements of business processes allocated to each group. It was found that, in practice, a particular CATool developed for a group should be further individualised to cater for specific end-user preferences and access restrictions.

For example in the case study CATool-2 was designed for “group-2” of the “machining cell”. This tool comprises software components required in respect to business entities BP-1 to BP-5, which relate to domain process 2, which previously was allocated to this group. In addition, CATool-2 was configured for end-users HNO1 and HNO2 separately as CATool-2A and CATool-2B. Using these configured versions of CATool-2, end-users can be assigned only limited access to the information system, i.e. access only to information entities required to carry out tasks allocated to them. Furthermore the properties of the graphical user interface

---

2 As mentioned before, ideally the CATools and their corresponding software components should be designed and implemented with respect to requirements of each enterprise activity, which in turn is defined by the semi-generic model. This can increase the configurability of the cell control software. However, in this case study (and this research) for the sake of manageability, generic software components were designed for each business process type.
(GUI) of each tool can be set up based on specific user preferences (although this facility was not implemented in this research). In addition to a configured set of software components, the CATools also comprise programming functions (§6.6), which enable each software tool to access the system infrastructure services and thereby interact with other systems outside the scope of the case study cell.

9) During this final step, pre-existing computer systems (i.e. their hardware and software) should be configured to provide the services required to allow the CATools to interoperate with supporting systems and cell physical resources. In this case, systems developed to support the case study work included a database management system, scheduler software, a LAN communication system, and a computer server. The CATools can run on individual computer hosts, connected to the server via the local area network (see figure 8.3). In this way the CATools can be distributed but still gain access to the information system so that they can receive defined schedules of activities such as job list, tool lists and other information required to complete a job. This approach also facilitates the specification of tools with a capability to collect real-time data from system elements for the purposes of monitoring and analysis (although this was not implemented in this study because of time constraints). The supervisor group also required access to the scheduling software to determine how tasks should be allocated to groups within the cell.

Regarding the design and operation of the supporting information system, cell data should be located in the database system (i.e. “core-data”, §7.5.2), and in the case of SQL commands addressing information (i.e. accessing information, §7.5.2) should be added to identify relevant information objects. During system operation these objects will be called as required by software components embedded within the CATools.

Although in this research, the use of an integration infrastructure (such as CIM-BIOSYS) is advised, for the case study work it was found to be sufficient and expedient to establish direct connections between system entities during system operation. This was achieved by using of network connections and management capabilities of a commercial database system – in this case Microsoft ODBC. However, in practice as the cell control system needs to expand to cover other parts of the manufacturing system it will become necessary to implement a suitable integration infrastructure to enable consistent access to the information system and to achieve flexible and scalable interoperation between system elements.

By using the modelling approach and model enactment toolset the case study system could be reconfigured and extended in scope with relative ease when compared with similar cell systems designed and built by conventional means. Inherent reconfiguration and extensibility capabilities can allow a system to be developed on an ongoing basis, even as requirements change unpredictably. Thereby improved performance of the cell should be enabled and the useful life
time of the system can be increased by organising, and therefore re-targeting the activities of people (i.e. engineering, managerial, supervisory and operator personnel) and machines within a target cell.

However, it was important to seek to quantify the extent to which system reconfiguration and extension can be enabled by the methods and tools developed during this research. Therefore the case study work was extended with the aim of assessing the extent to which system reconfiguration and extension can be enabled by the approach and to consider resultant benefits and practical constraints. However it should be pointed out that it was not practical to realise actual connections to (and therefore actual control of) the case study cell. Rather a pseudo-real target system was developed and run in the laboratory with modelled/simulated machines and personnel. However, it is known that the laboratory system could have been linked to real world cell resource elements by using well-established system integration approaches and mechanism.

8.3) Worked Example

Following a need to modify production methods deployed by the machining cell and its related cells, it was found that “hydraulic cylinder” production would require an additional inspection process to be carried out within the “quality control” cell. Initially for a given product type the quality control cell recorded an average of 2.5% part failures. This failure rate was deemed to be at an acceptable level based on current company policy. However for a short period of time after production, the supplier of the raw material experienced difficulty in maintaining the accuracy of the material size and quality. It followed that the part failure rate (especially within the machining cell) increased significantly and unacceptably to 12%. Furthermore, it was found that 75% of the parts that failed needed to be removed from the cell before further value added processing is carried out. To tackle this problem in the real environment, it was decided that two quality control units should be used within the “machining cell” to achieve “in-machining process” inspection to monitor report the results to the quality control department so that they can be analysed with respect to longer term trends via the use of statistical process control tools (SPC).

By using the modelling environment, implications of these new requirements and their impact on the “machining cell” were studied and candidate cell reconfigurations were developed accordingly. Figure 8.5 illustrates the design of the new layout of the “machining cell”, which was developed to implement the new quality control method. In practice reconfiguration of the case study model was achieved in a stepwise manner, as indicated by Table 8.4.

The model reconfiguration process illustrated by Table 8.4 shows the systematic way in which modelling of the case study cell can facilitate the re-engineering of the cell in response to unexpected change, which is outside of the original system scope (the change capability of the case study system will be further discussed in the next chapter).

In this example change scenario, the reconfiguration process can be enabled further via use of the
Use of this tool led to the generation of end-user applications (i.e. CATools) by accessing system configuration data that is specific to this particular cell in alignment with the required changes in configuration.

<table>
<thead>
<tr>
<th>Modelling Step</th>
<th>Change on Physical Cell System</th>
<th>Change on Modelling Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Intermediate inspection, Data should be collected and sent to SPC for control and analysing</td>
<td>Change on Business Entities, Adding DP-3 (inc. BP13&amp;14, EA6 to 10) (See Appendix F)</td>
</tr>
<tr>
<td>Procedural Rules</td>
<td>--</td>
<td>Generating the Allocated Procedural Rules for each inspection unit</td>
</tr>
<tr>
<td>Information Requirement</td>
<td>The quality control information and instructions are required for inspection processes.</td>
<td>Adding Information Enterprise Object EO-4 (inc. OV-41 to 44, IE 041 to 418) (see Appendix C)</td>
</tr>
<tr>
<td>Cell Resources</td>
<td>2 inspection operators (using manual devices)</td>
<td>Adding to human resources – HI1 &amp; HI2</td>
</tr>
<tr>
<td>Cell Functional Elements</td>
<td>Inspection operators should receive their &quot;job-list&quot; and report the control results in real-time.</td>
<td>Adding Cell Application Tools CATool-4 (A&amp;B) to be used by HI1 and HI2</td>
</tr>
<tr>
<td>Cell Grouping</td>
<td>Inspection group</td>
<td>Adding cell inspection group (group-4)</td>
</tr>
<tr>
<td>Group Allocation</td>
<td>--</td>
<td>Allocation the modelling entities to group-4, including: HI1 &amp;2, CATool-4 (A &amp; B), DP3 - BP13&amp;14, EA6 to 10</td>
</tr>
<tr>
<td>Generating Cell Control</td>
<td>Application tools (and a computer host) for each inspection operators</td>
<td>Regenerating all CATools based on the new modelling system configuration, Reconfiguring the network connects and accessing authorities.</td>
</tr>
</tbody>
</table>

Table 8.4: Modelling configurations required for accommodating the "machining cell" changes

8.4) System Reconfiguration and Extension – An Appraisal

The main aim of this research was to improve the reconfigurability and extensibility of manufacturing cell control systems by developing and using a system modelling and construction environment that facilitates (re)engineering processes when typical system changes occur. A requirement of this system modelling and construction environment was that it was capable of (re)engineering various types of cell, hence it was expected that the environment itself would need
to be configurable. However in this section of the thesis focus is on (re)engineering target cell systems not on re-engineering design and construction environments.

Various types of change with respect to the “Hydraulic Cylinder Machining Cell” were considered and the ability of the design and construction environment to accommodate those change types was studied. The change types considered were as follows: operational process change; physical resource change; production method change. Change to operational processes typically occurs in respect to manufacturing cells that are used to produce products or services in small to medium sized batches. Here operational details (e.g. size, quantity, machining process and tooling) may need to change even when the same class of product is being produced using a similar sets of resources. In the case study cell the particular model developed for the “machining cell” was found to be capable of accommodating this type of change by modifying the required information for the new processes and rescheduling the “job-list”. The provision of support for this class of change was considered earlier in section §7.4.

In addition, physical resources in a cell may need to change (or be reconfigured). This type of change is especially common when a semi-automated cell system is used to produce simple (possibly low quality) make-to-order products. Examples of this class of change are adding an extra machine or operator to the cell to perform a specific process (or processes) on semi-finished parts, or the unavailability of a machine because of maintenance procedures. A practical example observed in the company and considered here as case study was a minor product design change based on the customer orders. A “cylinder holder” was needed to be welded to the “hydraulic cylinders” for a limited number of products. These types of exceptional orders are handled by adding a welding process (and related welding machine and operator) to the cell, operated after the copy-turning processes. It was found that the particular model and its associated model enactment are able to be reconfigured to accommodate changes on the cell resources by regenerating the particular model for the new status of the cell. Consequent on physical resource change rescheduling of roles, tasks and activities will be required to achieve an appropriate reorganisation of interrelationships between resources. It was concluded that in the case study system changes of this type rely on the expertise of the cell supervisor (including knowledge of how to reschedule and balance the machining activities in the cell). In this case rescheduling knowledge was not formalised within the modelling structure. However if the supervisor requires a reorganisation of resources then the mechanism used to control, monitor and connect resources should be sufficiently flexible to perform a designated change.

Furthermore, the modelling system developed for the case study cell was found to be capable of accommodating production method change. The case study cell will in general require design modification to enable new production methods to be deployed, such as in the event of a need to manufacture a significantly different type of hydraulic cylinder. For example for a future yet-to-be-designed high precision part the grinding process may need to be replaced by or followed by a
honning process and this might also require the use of new inspection activities and support tools. Use of the modelling system should prove highly beneficial when this class of change occurs. An actual example of this type of change was discussed in section 8.3.

Summary
This chapter has considered the engineering of a case study cell, which partially tests practical capabilities of (1) semi-generic modelling concepts and (2) the modelling and model enactment environment developed in this study. Although the semi-generic model and model enactment tools were originally developed for use in PCB manufacturing domains, the test case was established in a metal cutting domain, thereby illustrating a reuse capability of the concepts developed during this study.

A “walk-through” modelling exercise in respect of a “machining cell” used to produce high-pressure hydraulic cylinders is described. Initially, the case study cell was modelled and operationalized as a prototype system, using concepts developed in this research. The resultant model describes the system structure and specifies aspects of the way that system implementation can be semi-automated to produce application tools (that support activities carried out by designated cell resources).

Three change classes associated with this type of “machining cell” (and indeed similar types of manufacturing cell) were considered to appraise the capabilities of the modelling concepts to accommodate change and therefore deliver advantages over contemporary design and build practice.

Further discussion and analysis of the case study results can be found in the next chapter.
Chapter

9

Analyses and Conclusions

9.0) Research Review

Current approaches and support systems used during the design and construction of manufacturing cells were reviewed. This illustrated the existence of a gap between available design and construction tools that is known to result in an inability to readily change the functional capabilities of manufacturing cells. This was taken as the justification for pursuing further research on the design and configuration of manufacturing cells.

Initially this research was broadly aimed at improving the “change capability” of manufacturing cells. However, early project study focused activities on developing an integrated enterprise modelling and model enactment environment that facilitates the design and construction of customised computer control systems from reusable control system components. The following main research objectives were set (§3.3).

- To specify and develop ways of using a semi-generic modelling structure to support the explicit representation of manufacturing requirements and design specifications for particular manufacturing cells.

- To specify, develop and test the use of a methodology that lends structure to the process of designing manufacturing cells and constructing manufacturing cell control systems from modular elements.

- To develop methods and techniques that take as an input particular models of manufacturing cells and generate as an output an appropriate configuration of reusable software components, so as to operationalize modular functionality specified by the semi-generic model.

In support of the above the study has had secondary objectives, namely: a) to define, implement and test the use of a generic information structure capable of underpinning the modelling of manufacturing cells; b) to specify and use a set of prototype infrastructure services capable of underpinning the life cycle of manufacturing cells; c) to prototype the development of a set of software application tools that enable the practicability of the research approach to be assessed.

The research was mainly concentrated on developing methods that can improve “change capability” of manufacturing cell systems. To achieve this, it was initially envisaged (§4.1) that the resultant modelling system should be “reconfigurable”, “reusable”, and defined in a “generic” manner.

A set of modelling and system integration concepts that had previously been developed as part of pre-existing enterprise modelling approaches were selected and their combined use was organised
and developed into an enterprise modelling and model enactment environment.

9.1) Discussion of Research Achievements
The primary research achievements that have added to the body of previous knowledge in the field are classified as follows: a) a consistent set of systems engineering concepts, capable of underpinning the application of a component-based approach to the design and configuration of manufacturing cells; b) a methodology based on the reuse of a semi-generic model that lends structure to the use of the set of systems engineering concepts in two fairly distinct manufacturing domains; c) a set of semi-generic modular building blocks (or system entities) that can be aggregated to construct and interconnect modelled and real cells in a given manufacturing domain; d) a prototype set of modelling and model enactment tools that enable system designers and builders to utilise the concepts, methodology and modular building blocks.

The Concepts — Prime focus of attention was on unifying the use of two largely distinct concept sets, concerned with (1) the high-level modelling of cells, mainly in terms of their functional and informational requirements and capabilities, and (2) component-based system design and construction approaches that focus on the reuse of software building blocks. Many high level modelling concepts and modelling structures have been described in the literature. Although these have significant potential, their use by industry and academia has been limited thus far. An assumption made in this study is that much of the unrealised potential can be unlocked by (A) retaining and reusing knowledge contained within design models about how modular components need to interact in order for them to realise an explicitly defined set of requirements, and (B) reusing aspects of that knowledge in the form of computer executable models that can function as part of real cell control systems. This notion has been developed and part tested during this study and has been shown to improve the change capability of cell solutions. Here it has been shown that knowledge reuse can repay investment made in modelling by enabling systems to operate in close alignment with requirements, even when certain aspects of those requirements change in an uncertain manner. In this research the use of a coherent set of modelling and model enactment concepts was structured via the use of a semi-generic modelling structure. This targeted system development on the reuse of “classes” of system entities (that were not simply a collection of predefined system instances). Furthermore, the descriptions of system entity classes are believed to have the potential to be used by different vendors to guide their future development of reusable software components.

Methodology — The CIMOSA framework was chosen as the backbone structure for enterprise modelling. This was found to be suitable for use in defining system requirements in informational and functional terms. However, to achieve the research objectives of connecting modelling and real environments (in terms of models of cells, configured software modules and physical resources in a particular cell) it was necessary to conceive, specify, develop and test a
so called Resource, Allocation and Configuration Method (RACM). The RACM structures the selection and configuration of reusable building blocks of cells (i.e. system components) and assigns to component groupings sets of generic “tasks”. The building blocks define: physical resources required to carry out tasks; human resources, to perform supervisory tasks; supportive tools and infrastructure services; and interaction techniques needed to establish communication with other system elements and other systems. Building block definitions led to recommendations on how actual software components should be specified in order to enact (or operationalize) models of building blocks. A suitable set of software components (along with administrative and service functions) were defined and developed to illustrate how cell control application tools can be constructed for a specific cell system.

The RACM approach proved to be effective during the study but it was also found to be dependent on (a) the completeness with which system requirements can be specified and (b) the capability of the software architecture selected to enable the system components to interoperate.

In addition, a method was developed that structures the capture of semi-generic information requirements and transforms these models into a computer executable form. Here CIMOSA information objects (and associated sub-objects) were redefined as equivalent EXPRESS ‘schemata’ (and related sub-entities). The EXPRESS model so developed provided a semi-generic information model of manufacturing cells used during PCB manufacture. Subsequently, an information system was designed and developed based on this information model and used to underpin the development and operation of specific cell controllers.

Generic system entities as Modular Building Blocks of Manufacturing Cells – Various types of generic and reusable cell system entity were defined and their use developed. Initially these entities were defined to improve the change capability of manufacturing cells used during PCB manufacture. However experience gained from the case study confirmed that these entities can also be reused in another domain.

The generic entities defined include: a) generic tasks, found in PCB manufacturing cells that were formally structured as set of enterprise activities, their relationships and operational rules; b) generic functional entities, with functional capabilities to perform generic tasks that model physical resources for reuse within the modelling environment; c) generic information entities, formally defined as information elements and related entities; d) system components (and associated software components) that can be reused when generating cell control applications.

Prototype tools – To facilitate the demonstration and evaluation of the research concepts, two types of prototype tool were developed, namely a “Cell Design Tool” and “Cell Application Tool”. The CD Tool facilitates use of the semi-generic model in particular cells. This tool semi-automates the generation of CATools in a manner that is structured by a particular selection of system components. Sample software component building blocks were also developed to enable
CATools to be constructed and tested.

The tools were designed for use in proof-of-concept experiments. However it was found that they would need to be redesigned and implemented to facilitate their real world application.

9.1.1) Analysing the Case Study Results

An industrial case study evaluation of the project concepts was carried out in a metal cutting domain, whereas the original concepts were conceived with reference to common cell-control system requirements at a printed circuit board manufacturing plant. The project methodology and modelling structure was applied to an existing “AS-IS” cell. This led to the development, of a particular model of an industrial cell, which was used to facilitate the selection, configuration, implementation and operation of a component-based control system. Subsequently, to illustrate the change capability of the project methodology and modelling structure, an actual case of change was catered for. This evaluation work demonstrated an enhanced change capability over and above current practice. Implications of this enhancement are discussed further in the next sub-section of this thesis.

Although it was not practical to carry out the change capability evaluation work at the case study site, the models developed were populated with real-case data. Also the operation of physical cell resources was simulated. Therefore it was assumed that the study results represent a good reflection of the enhanced capability of the project methodology and modelling structure to address real cell control configuration and change problems, as typified by the case study cell.

The evaluation work also demonstrated the capability of the project concepts to target and support the efforts of designers and constructors of cell controllers by providing a systematic approach to modelling existing manufacturing cells, as demonstrated by the “hydraulic cylinder machining” cell. It also demonstrated how the modelling approach can be incrementally implemented within an existing manufacturing cell. Figure 8.2 (i.e. context diagram) shows that the modelling system was applied only to a single manufacturing cell (i.e. “Machining Cell”) that was defined as a CIMOSA-conformant domain. However, use of the approach and structure could have readily been extended to include other cells on the shop floor of the case study company. This showed that in principle the approach and modelling structure was scalable.

In addition to the possibility of applying the modelling approach to some part of the manufacturing cells, it was also shown by the case study that the semi-generic model can be partially applied to the systems. This was shown by adopting 2 domain processes (from 3 predefined generic DP’s) as indicated by figure 8.2 (i.e. structure diagrams).

As part of the case study work, the way that a semi-generic model can be used to generate particular models of a specific cell was investigated. Table 8.2 summarised the findings and describes in outline how functional entities (FE’s) were selected and instantiated from equivalent
types of generic entity.

Also examined during the case study was the process of reconfiguring models when cell conditions were changed. Table 8.4 characterises the process steps followed in order to accommodate for the inclusion of an additional quality control process and an associated SPC system within the cell.

The case study work also demonstrated the capability of the approach and modelling structure to target cell system design and construction on the use of a predefined set of software tools and services. It also showed that the approach and modelling structure was sufficiently flexible that it could retarget design and construction on alternative integration services. For example use of a particular LAN (rather than CIM-BIOSYS) was adopted for the case study work, as was the SPC application software. Furthermore, it was proven possible to retarget use of the modelling structure towards the use of other existing standard infrastructure services and tools. Longer term, however, a requirement was identified to target use of the design and construction approach on standard infrastructure services and tools used industrially.

In summary, the industrial case study demonstrated that improved change capability can be practically achieved by deploying the cell system design and construction concepts specified and developed in this study. However, during the evaluation work, change to the actual runtime operation of the case study cell could not be made in situ, nor could access to actual machines be realised in the laboratory. Hence changes to runtime operation were only partly demonstrated and proven. Furthermore, had further project time and resource been available it would have been highly beneficial to investigate other ways of developing control system components, at various levels of granularity. It would, for example, have proven interesting and beneficial to target cell operation on the use of other (standard) infrastructure services and supporting tools.

9.2) Analysis of the Change Capability
This section will consider further "change capability" attributes of the design and construction approach and modelling structure. Analysis is made with respect to the "model of change" described in chapter 1.

Figure 9.1 depicts common types of manufacturing system change considered. Focus of attention has been on a study of the implications of these change types on function and information requirements in two domains of manufacturing cell systems. The initial domain studied related to various cells used to manufacture PCBs, whilst focus of attention during the concept evaluation work was on a particular type of metal cutting cell. This figure also indicates how the main parts of the modelling environment were found to be capable of supporting different types of system reconfiguration as requirements changed.

Generally three broad classes of change were studied in this research, namely: (1) change to business requirements, (2) change to production requirements, and (3) change to operational
behaviours.

1) **Change to Business Requirements** – This type of change, which will be referred to as type (1), includes major changes to enterprise definitions, such as might occur when defining new goals and organisational structure on a wide scale, or when a major change is made to production, engineering or business methods. This type of change might alter basic assumptions used when developing the semi-generic model. Although covering this type of change was not the main concern of this research, it was envisaged that change of type (1) might be supported to some extent. However, to provide comprehensive modelling support for this type of change, it was found that significant additional thinking and supporting concepts would be required in a given practical situation to reselect, redefine and reconfigure appropriate semi-generic model elements in terms of tasks, system functionality requirements, system information requirements, and various interrelationships.

Experience from developing and using the project concepts, component-based approach and modelling structure led to the following conclusions about their capability to support change of type (1).

- **Business Change** – A “business model” of a given enterprise system could be developed to describe the essence of enterprise processes and possibly the impact of various changes. Models of manufacturing cells developed during this research could be linked to such a

![Diagram](image-url)

**Figure 9.1:** Part of the common changes in manufacturing system studied in this research
business model. This could help to specify (and modify) assumptions made about manufacturing cell system requirements. The semi-generic model of cells (used during conceptual design and represented by a 'context diagram' – see figure 5.4) can be influenced by business change. The impact of some of these changes can be linked to models of manufacturing cells by extending the domain of the modelling system and defining relationships that exist between the focused area of modelling and other parts of the manufacturing enterprise. Examples of linked changes might include: producing a new range of products; focusing on low quantity orders (rather than mass producing); reducing stock levels; introducing new definitions of responsibilities and authorities of personnel; etc.

- **System Engineering Changes** – A mechanical design change, production management strategy change, or a production planning change might well impact on system engineering changes. Depending on the magnitude of the change, a major modification or reconfiguration may be needed to a manufacturing cell when such changes occur. Although time constraints did not permit testing of the modelling environment with respect to changes of this type, it was assumed that modelling support might not be able to accommodate most consequences of engineering change. For example, a manufacturing business system might consider the use of various production strategies such as traditional manufacturing control systems, computer integrated manufacturing systems (CIM), or new generation of manufacturing control systems. However, in this study the effectiveness of modelling support environments was only considered within the context of a CIM environment. Another example of engineering change might follow the introduction of a production planning and control system (such as an ERP system) that might result in new strategies used to plan and control manufacturing systems.

To summarise, it was concluded that change of type (1) could be partially supported by approaches developed in this study. Models can be reconfigured (or redefined) during “conceptual design” by modifying generic business entities (i.e. DP’s, BP’s, EA’s), generic function entities and their relationships. However, if the impact of type (1) change cannot readily be mapped into change to business entities, major re-engineering work may be required at the conceptual level to link cell and other enterprise systems in terms of the impact of change. It was understood that type (1) change could impact on choice of generic methods (and consequently generic system components) used during the “detailed design” of cells. Indeed generic methods may need to be re-engineered specifically based on details of the new requirements and specifications.

---

1 Examples of new manufacturing control systems and concepts are agent-based manufacturing systems [113, 163], Holonic concept [48, 164], and Bionic concept [164].

2 Examples of alternative strategies to plan and control manufacturing systems are the use of JIT [165] and OPT [11].
2) Change in Production Requirements – This type of change will be termed type (2) and has been the main concern of this research. Change of type (2) has been assumed to include change to the way that a product is manufactured. Typical impacts of type (2) change are: the use of different types of cell resource and resource configurations (e.g. functional, process layout or specialised cell groups); use of different supporting tools (e.g. remote scheduler, local dispatcher, SPC systems [166], central databases [167]); the use of different physical resources; use of different operational policies (e.g. change to scheduling, ordering and inventory rules, tooling, quality control and cell monitoring policies); and so forth. Change of type (2) is typically linked to a need to define: a) new tasks allocated to a cell system, b) a new selection and organisation of resources required to perform the new tasks, c) a new set of infrastructure services to support operations related to the new tasks.

To accommodate change of type (2), the particular model of a cell has to be redefined, to match the new requirements. It was found that “production requirements” change typically has an unpredictable nature and in general cannot be pre-programmed when designing and building any given cell system. However, this study has shown that modelling systems and component-based cells can provide a degree of “change capability” (i.e. reconfigurability) that can meet a range of production requirements change. This research has shown that a prime source of “reconfigurability” arises from reuse of the semi-generic model and the prototype application tools. This reuse property facilitates the redefinition and realisation of new versions of a particular model that meets new (not previously anticipated) production requirements and conditions of a cell system.

Change of type (2) is mainly supported by the semi-generic model at the “detailed design” level of modelling. Typically it results in changes to: operational processes (§7.6); resource configurations (§6.5.3), network connections (§6.5.7, §7.5.1), information structures and entities (§7.5.2), as discussed below:

- **Process Change** - When a new process is defined (in order to produce a different part or to modify an existing process) the modelling structure needs to be reconfigured to accommodate the change. The change can have two effects on the modelling system. Firstly, enterprise activities, that are allocated to “cell groups”, need to be reconfigured. For instance, in a PCB production line a drying process may be added after soldering and washing processes. Or in a metal cutting scenario a new inspection system may be added, as illustrated by the case study. Secondly, the new process model needs to be populated with real data (e.g. drying part X by machine Y at time T, etc.). This is carried out by detailing particular models. Having defined a new process the flow of operations must be rearranged. This can be achieved in the modelling environment by modifying the “allocated procedural rules” (§6.5.6).

- **Resource Change** - Change to process plans may require a new class of physical device or
human resource to be defined along with new supporting services. Such changes may also
demand resource reconfiguration. Typical examples are: the replacement of a broaching
machine with an EDM machine, adding an operator confirmation to continue a particular
operation, providing local scheduler software for cell supervisors, accessing a new part of a
database within an information system. The project modelling environment supported this type
of change by facilitating modification and reallocation of resources to the cell groups.
Additionally it was found that resource specifications may need to change and hence
descriptions of generic system components and possibly software components (defined during
“detailed design”) may need to be modified.

- Change in Information Requirements – Any modification to processes, procedural rules,
resources or services necessitates preparation and updating of supporting information structures
and entities. Operational changes will typically be linked to a modification of the design of the
semi-generic information model. Indeed modification to information objects views (OV’s) and
the information elements (IE’s) may be required. Subsequently, the information model needs to
be executed to relay the impact of changes to the information system. However, the main
structures of the information model (e.g. its three schemata, database interactions, data
exchange methods and format and certain conceptual information requirements) will typically
remain unchanged.

- Integration and Networking Systems Change – Engineering changes may also require
modification to supporting integration services and the networking system. The “detailed
design” model defines the methods implemented for these supporting elements. Examples of
typical support system changes are the replacement of the CIMBIOSYS integration system by
alternative systems3.

3) Change in Operational Details – This type of change will be referred to as type (3) and
corresponds to change to operational instances of tasks allocated to a cell. Type (3) is typified by a
job change and may require modification to both operational processes and use of infrastructure
services. In this research, type (3) change has been considered to be a specific subdivision of
change type (2).

Examples of change type (3) may arise because of: changing the physical data in a process plan
(e.g. part quantities, due-dates, priorities); re-populating database tables, based on new cell status;
using different protocols for integration and data exchange; generating new work-to-lists to
produce different parts from a common family with different quantity, size, tolerances, etc. (within
device capability limits).

---

3 Examples of alternative integration systems are localised network system (e.g. LAN [168], as used in the case study), CORBA [109] and Internet-based interaction systems [161].
It was found that change of type (3) is typically of a programmable nature that can best be catered for during the runtime execution of cell systems (see figure 9.1). Typically it is also expected to occur more frequently than type (1) or type (2) change. However to be able to accommodate this type of change within the project modelling environment, certain modifications to system resources may be required to be utilised at the implementation level of modelling a particular cell.

This research did not seek to investigate implications of this type of change during the run-time execution of cell systems. However, it was concluded that the modelling system can be used to facilitate change of type (3) and thereby to associate this change in other external systems, e.g. when rescheduling a process plan.

Table 9.1 summarises the extent to which “change capability” is supported by the modelling environment developed as part of this research.
<table>
<thead>
<tr>
<th>Change Category</th>
<th>Example of Changes</th>
<th>Support provided by modelling system developed in this research</th>
<th>Prototype tools developed and used</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type (1)</strong> Business Requirements</td>
<td>High-level business changes, such as: Management methods, Marketing goals</td>
<td>Supported</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>System engineering changes, such as: Mechanical design changes, Production mgmt strategy, Production planning and control system</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Type (2)</strong> Production Requirements</td>
<td>new cell grouping arrangement, new tasks allocated to cell groups, different set of physical resources, different external tools (e.g. SPC), new human resources, new information / integration system configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tested and industrial practicability partly proven</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Type (3)</strong> Operational Details</td>
<td>new operations within a task</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>new scheduled list (e.g. work-to-list)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>new data access/exchange method</td>
<td>tested</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 9.1: Summary of research achievements aimed at improving “change capability”.

163
9.3) Final Discussion and Conclusions

Here general conclusions about the research are drawn. It is concluded that initial research objectives were achieved when viewed from a complementary set of perspectives. However, shortcomings of the research were also identified and discussed below.

**Improvement in system reconfigurability** - In this research, the “reconfigurability” was interpreted as being the ability to reorganise and adjust system elements and their relationships in a systematic manner to enable the overall system to achieve a common purpose or goal.

In this research a set of methods and techniques were proposed and developed that design and construct manufacturing cells from semi-generic and reusable system elements. Requirements and specifications of these system elements were defined by the semi-generic model, and relationships amongst them were realised by “system components” (and associated software components). Therefore in the event of system requirements change, system elements and their relationships can be readily changed and adjusted.

In general, based on research findings it is concluded that improvements over current practice reported in the literature can be made by deploying the design and construction of methods and their supporting modelling structure and thereby lead to the development of “reconfigurable” and “change capable” manufacturing cells.

**Reusability** – In this research a set of semi-generic system entities were defined and their use was tested. They were found to be reusable when developing cell control systems for a specific domain of manufacturing cells. The system entities included: “generic tasks”, that define functional requirements of a cell system in the form of business entities; generic function entities, required to perform a class of tasks; information objects required for to perform tasks; “system components”, that define a reusable building block comprising information objects, procedural rules, function entities, infrastructure services, and their relationships, required to perform a particular class of task formatted as a business process. A set of prototype “software components” were also developed, tested and reused when implementing “system components” at the operational level.

Furthermore the semi-generic system entities were found to be reusable (without any major modification) in a new manufacturing domain, namely a common class of metal cutting cell.

Also with regard to the modular structure of the modelling system and its inherent ability to reuse system entities predefined by the semi-generic model, it was concluded that the resultant modelling environment should be applicable for the design and construction of many types of manufacturing cell and in various industrial domains.

**Generality** – One of the fundamental objectives of this research was to develop a modelling environment able to facilitate the design and construction of cells belonging to a “domain” of
manufacturing. The PCB manufacturing cell domain was initially targeted and requirements for various PCB cells were used to develop the modelling system. However, use of the modelling system was also tested in a machining cell domain (see case study).

An investigation of the practicality of using a semi-generic model to support design and construction has been an important aspect of this research and on the basis of the research achievements and the case study results, it is concluded that new findings about the reuse of such models have been generated.

**Expandability** – The developed modelling support system has the capability to support some phases of the life cycle of cell controllers. However to provide more complete coverage, work in this study can be extended by a) linking the modelling domain to other modelling domains (e.g. concerned with high-level business modelling and low-level operational system engineering), or b) connecting semi-generic model entities to other enterprise entities to map the impact of different types of change.

It is probable that coverage of the modelling support system could be extended to include improved modelling from a resource view and linking this to improved modelling of organisation, human and financial issues. However, it was considered that this would best be achieved by enabling the models of cells to communicate with other models contained in other modelling environments. An example of such a resource modelling system was developed by Li [35] and a candidate operational modelling system was proposed by Aguirre [162]).

The use of standard methods and the ‘openness’ of the semi-generic modelling system (being based as far as possible on CIMOSA modelling constructs) means that the modelling support system should be readily expandable. In fact primarily studies were carried out by the author to investigate the possibility of integrating the modelling system developed during this research with that of another modelling system developed by the University of Navarra in Spain (see section Future Work).

**Rapid reconfiguration** - To cope with fast-changing manufacturing markets, a modelling support system and the systems it is used to design and construct should be easy, fast and cost effective to change.

This research has developed component-based concepts leading to the predefinition and reuse of a set of system components. Their reuse can reduce the time and cost involved in system design and build processes. Furthermore, the ‘openness’ of the developed modelling system should allow designers to use it in conjunction with other modelling environments and cell products. Therefore it is concluded that the use of modelling support system developed in this research can reduce the time (and cost) required to design and reconfigure cell systems.

**Practicability** - It was not feasible (or possible) to fully assess the industrial practicability of the
semi-generic modelling concepts, developed and used in this research. Firstly because there is a link between the development of the semi-generic modelling system and the development of reusable system entities (i.e. particular models and cell application tools). The role of semi-generic models was to guide system developers and component vendors and to recommend system entity descriptions that can lead to the development of particular models and cell control products by using various methods and techniques (including the methods proposed in this research). Here the practicality of using these system entities as commercial products in actual industrial cases also needs to be assessed. A much more wide ranging set of evaluation exercises might actually prove or bring into doubt the practicability of reusable system entities and particular models for industry from a semi-generic model. Even then it is very unlikely to rule out the value of the general guidance and assistance offered by this kind of modelling approach.

Secondly, there are many complicated factors involved in engineering manufacturing systems that cannot all be considered when designing a prototype modelling support system. Evidently when a laboratory-tested support system is applied based on a real case study, many simplifying assumptions are made when developing a working environment similar to the conditions assumed during the design phase. Some of these assumptions and their likely impact will be more evident and easy to quantify than others. Hence to obtain an accurate assessment, the support system should be used for long periods of time to fully appreciate the impact of simplified (or ignored) factors on the support systems. Clearly a full analysis of such factors might create a new view of system requirements and identify opportunities to improve the prototype modelling support system based on an aggregation of real industrial needs. It is understood that developing a test-bed to evaluate a manufacturing support system under special circumstances can neither strongly validate nor reject concepts embedded into the modelling system.

Thirdly, there is much literature reporting on the successful industrial utilisation of modelling architectures, such as CIMOSA (see Appendix G - §G.4.1). The majority of these reports imply the use of modelling methods to tackle specific (and mainly operational) issues of systems in a restricted domain (i.e. development and use of particular models). However there are very few examples of the use of methods in which particular models have been altered (i.e. reconfigure) to meet changed requirements, except in some cases the development of a 'reference model' for a particular domain has been mentioned. Whereas in this research, the modelling support system has added a reconfiguration capability via its means of utilising semi-generic models to change particular configurations of cell system entities. Therefore to some extent it can be said that the practicability of underlying modelling concepts used in this research (i.e. CIMOSA based concepts) has already been fully tested. Here the 'products' of the semi-generic model (i.e. particular models and cell control tools) have been tested. However, it is also important to test
the methods of generating those products.

In general, the author believes that although the modelling system developed in this research has not (yet) been completely implemented in real industrial cases, it does offer design principles that can facilitate the development of new design methods and the development of systems that can be practically applied industrially. However, a number of difficulties (in terms of development of system structure and design approach) can be anticipated based on the results of the industrial implementation of the modelling systems. As these and other difficulties are identified it is expected that the modelling methodology will need to be revised to tackle those difficulties.

Improve the agility – What constitutes improving the agility of a manufacturing system is a highly complex subject. There are many complex issues involved in the design and construction of manufacturing systems that influence the agility of the overall system. Improving each issue might well increment design methods toward desired levels of agility. This research was focused on improving the reconfigurability of manufacturing cell systems. Therefore it is concluded that the modelling system developed in this research has contributed an incremental advancement in the agility of manufacturing systems. However the significance of this advancement cannot be measured based on the research achievements of this study alone.

Choice of research approach - The applicability of a research approach can be approved or otherwise by evaluating whether a) the research objectives have been achieved by the approach taken in a given time, b) further research can be built upon the results of that research. However, the method of evaluation cannot assess the quality of the research approach, i.e. if the approach was a good, effective or simply a feasible approach.

Here it is concluded that a significant fraction of the research objectives were achieved (see §9.1 and Table 9.2) and also that the research has delivered results that can be used by other researchers to expand the research domain. Therefore the approach taken in this research can be considered to be appropriate. However, by no means it is believed that the research approach proposed is the best possible one that could be taken to solve cell design and construction problems. There are many shortcomings and difficulties that will arise from implementing the concepts developed based on this research approach (these shortcomings will be discussed further later in this section).

Contribution to the state-of-the-art - The idea of developing a generic modelling structure has been previously investigated by many researchers. Nevertheless, apart from research projects that have proposed use of the generic modelling level as part of their modelling architecture, there are few researchers that have actually expanded their concepts to facilitate manufacturing modelling construction in such a way that it can link to a higher-level of modelling. Other researchers have typically referred to higher-level models as simply 'reference models'.
Whereas, as discussed earlier (§4.5), this research distinguishes reference models from generic models (generic models can include reference models, however).

On reviewing the literature, the author is not aware of other research work in the area of modelling at the generic level that actually proposes the use of formalised methods to develop and implement generic models and thereby to assist in the generation and configuration of detailed models (i.e. particular models). Therefore in this respect, this study has contributed to advancement in the state-of-the-art by studying and formalising the use of the CIMOSA modelling architecture at the semi-generic (partial) level of modelling.

In addition, this research has proposed and structured the use of component-based concepts in associated with existing enterprise modelling approaches (here, the CIMOSA architecture) to develop design methods based on modelling. It is also able to improve the modularity and reusability of system designs.

Furthermore, within the research framework, a set of generally defined modelling entities were developed and tested and proved to be reusable in two manufacturing cell domains. Moreover, the definition of generic modelling entities was extended to define precepts of software components that can be generated into a practical and executable form by various commercial application builders. Thereby software components can perform actual operational activities associated with generic entities (e.g. software needed to perform a generic task).

Here it can be concluded that some improvement was made in bridging the gap between top-down modelling approaches (applied to manufacturing cell systems as provided by classic modelling methods, e.g. CIMOSA) and bottom-up cell control solutions, mainly offered by commercial vendors.

**Research Weaknesses** – A number of shortcomings of the research approach were identified that could not be tackled, primarily because of time limitations. The major conceptual difficulties identified are outlined below.

- The research sought to develop a modelling support system to cover the life cycle of cells, from conceptualisation, through design and implementation to the provision of some support capability during the runtime operation of cells. However during the research, it became evident that it would be necessary to limit that coverage, particularly with respect to it not being able to support the implementation of and change to actual electro-mechanical resources in cells.

- The rational for selecting the CIMOSA concepts for aspects of the modelling structure (i.e. during requirements capture and conceptual design) was explained in section 2.7. Although subsequent research findings confirmed the appropriateness of this choice, the author believes that more extensive study and experimentation aimed at testing the use of other modelling
approaches might have led to an improved selection of concepts, methods and tools.

- The research study was initiated by developing a custom-designed, bottom-up approach to the design of the manufacturing cell control systems. This targeted the subsequent development of top-down design and build concepts. Hence subsequent research was mainly concerned with theoretical issues leading to a suitable modelling system. However throughout it was understood that a closer contact with industry might have reshaped the final form of the design and construction approach and the final choice of reusable system entities.

A number of technical difficulties were also identified as follows.

- The research developed a formal way of describing manufacturing cell controllers from requirements, through conceptual design to a detailed definition of configured elements of cells. Although the formalisms used proved to be appropriate, a need was identified to formally model the flow of the activities at the implementation level, hence the use of a formal language might have been advantageous. Languages such as Petri-Nets\cite{103,157} and Estelle \cite{156} could have been utilised to precisely formalise the flow of activities and events and "connect" these flows to resource functionality, possibly by defining and describing more completely the procedural rules.

- As engineering methods and IT concepts have advanced significantly during the last few years, some of the methods and techniques used in this research could be viewed as being outdated and the choices made may require revising. For instance new enterprise modelling environment could have been considered, such as GERAM \cite{62} modelling architecture, and the CIMBIOSYS infrastructure could have been replaced by the use of Java and Internet-based integration services.

- The research does not provide a clear definition of a method for modifying the information system and therefore for updating the data storage system when system change occurs.

Table 9.2 summarises the research achievements.

9.4) Recommended Future Work

- Modelling support could be usefully extended in two directions. With the purpose of placing the engineering of manufacturing cells within a wider context, the CIMOSA requirement definition modelling level could be linked to other business modelling systems. Ongoing research projects such as KARE \cite{112} and SADRES \cite{169} provide concepts and mechanisms for capturing and formalising a wider range of requirements. Hence these modelling methods could be used to specify key elements of the requirement definition model of cells. Whereas with the aim of improving implementation aspects of the support environment, it is likely that

---

\footnote{Petri-Nets has been widely used by the SEWOSA CASE tool and was experimented at the first stages of this research (Appendix E).}
additional concepts could be identified and used that have been developed within other research studies focused on operational aspects of cells. For example it might prove practical to enable the modelling environment to instruct the execution of the physical system. An example of this kind of research is that carried out by Aguirre [162] at the University of Navarra in Spain. This opportunity has stimulated collaborative work between MSI and the University of Navarra, as reported in Appendix H.

- When carrying out reengineering processes in response to required system change, significant modelling effort (both manual and automated) is required to reconfigure particular models and regenerate CATools. Hence a simulation environment could have been added into the modelling structure and environment (as a supportive tool) to facilitate analysis of results during reengineering and before seeking to implement the reconfigured processes. Work reported by Reithofcr [170] and referred to as the Virtual Factory Lab Victor could provide concepts to support the development of such a capability.

- Regarding the generation of the software components, additional end-user preferences could be added to descriptions of components to provide a more user-oriented, model execution system. Reason for customisation and ways of developing a model execution system (based on the user preferences) was investigated by a ROPA project [161] within MSI. The author was employed as a research associate on this study, which followed completion of the research reported in this thesis.

- By developing complementary bottom-up approaches, other semi-generic models and system entities could be developed for other domains of manufacturing cells. Thereby the modelling structure could be tested and used with respect to other kinds of manufacturing cell and could lead more broadly to improved usability of generic shop floor entities. This kind of work could be linked to the development of a comprehensive reference model of domain-specific cell system entities.
<table>
<thead>
<tr>
<th>Research Objectives</th>
<th>A semi-generic model to represent requirements of a domain of manufacturing cells</th>
<th>Developed and tested from information and function viewpoints, and its use partially proven via case study work</th>
<th>Modelling scope can be extended to cover high-level business requirements as well as low-level operational requirements</th>
<th>Developed based on one industrial case. Model validation is expected to require testing in several cells in the same domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>A methodology to facilitate design of manufacturing cells and cell control systems</td>
<td>RACM approach developed (that can be driven by SEWOSA models) and used and its potential benefits analysed</td>
<td>RACM is based on a set of assumptions about bottom-up development of TO-BE cell systems. However it is assumed that a number of external tools (e.g. simulation tools) may be required to effectively apply this method and improve AS-IS systems.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component-based software capabilities to enable use of the models and design methodology at the operational level</td>
<td>Specification of a set of semi-generic software components was defined and a set of prototype software components developed</td>
<td>Generation and allocation of software components, the reuse of which was tested under laboratory conditions – however subsequent collaboration by software vendors could lead to more practical sets of generic and customisable components</td>
<td>The software architecture needs to formalise development, selection and modification of software components</td>
<td></td>
</tr>
<tr>
<td>A generic information model for the given cell domain</td>
<td>Semi-generic information requirements were defined, an information model was designed and an information system implemented for laboratory use</td>
<td>With the use of appropriate commercial applications, it is assumed that the semi-generic information model can be practically used in industrial systems</td>
<td>Information aspects of “change capability manufacturing cells were not sufficiently investigated</td>
<td></td>
</tr>
<tr>
<td>Prototype infrastructure services for the modelling and real time execution systems</td>
<td>Prototype information and integration systems were developed and tested to underpin modelling and operational phases of manufacturing cells</td>
<td>If the prototype integration system (and its interaction and data exchange means) used in this research is to be used industrially, major enhancements will be required</td>
<td>The prototype integration system (interaction and data exchange means) used in this research is not expected to be industrially efficient and usable</td>
<td></td>
</tr>
<tr>
<td>Change Capability</td>
<td>Change to operational requirements (type 2) was tested and proved</td>
<td>Change to operational details (type 3) should be supported but business requirement changes (type 1) was a prime aim of this study</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reconfigurability</td>
<td>Improvements made and demonstrated</td>
<td>Should also be effective for AS-IS systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reusability</td>
<td>Reusable system entities were proposed and used when building cell systems</td>
<td>Cell design and construction should be faster, cheaper and easier</td>
<td>The customisation of “reusable system entities” was not studied</td>
<td></td>
</tr>
<tr>
<td>Being generic</td>
<td>Modelling support was developed for use at the semi-generic level</td>
<td>Should be applicable in similar domains of manufacturing cell</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expandability</td>
<td></td>
<td>Because standard methods were used and also because CIMOSA models are expandable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Practicability</td>
<td>Theoretically proven in support of the test case</td>
<td>Assumed to be practical and beneficial for real cells</td>
<td>Was not tested industrially</td>
<td></td>
</tr>
<tr>
<td>Contribution to the state-of-art</td>
<td>Extending and formalising the use of the CIMOSA approach at a partial level of modelling cells, defining links between enterprise modelling approaches and component-based software technologies, Provision of a solution to bridge the gap between bottom-up and top-down approaches to the design of manufacturing cell systems</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9.2: Summary of the research result
Appendix

Library of Semi-Generic Building Blocks

This appendix documents sample sets of generic building blocks defined by this research. It also categorise these blocks as a library of reusable system entities (see §6.6). Five classes of reusable entities are discussed below, namely: business entities; function entities; information entities; procedural rules; and software components. These entities should be used by system designers to construct particular models and to generate end-user cell application tools (CA Tools).

A.1) Reusable Business Entities

Cell tasks were defined and structured (§5.3.1 and Appendix F) into the form of business entities comprising domain processes, business processes and enterprise activities (§G.4.1). Generic tasks defined in respect to the PCB domain of manufacturing cells are outlined in table A.1.

Table A.1: Library of the Building Blocks - Reusable Business Entities

<table>
<thead>
<tr>
<th>DP</th>
<th>BP</th>
<th>EA</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept New Orders</td>
<td>10012</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reject New Orders</td>
<td>10013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive New Orders</td>
<td>10101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Cell Capability</td>
<td>10102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Eng. Data</td>
<td>10103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Due Date</td>
<td>10104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirm Cell Capability</td>
<td>10114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collect Static Data</td>
<td>10505</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collect Dynamic Data</td>
<td>10507</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedule Tasks</td>
<td>10606</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedule New Material</td>
<td>10608</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedule Tools</td>
<td>10609</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedule Equipment</td>
<td>10610</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collect Static Data</td>
<td>10505</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collect Dynamic Data</td>
<td>10507</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provide Work-To-List</td>
<td>10411</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiving WTL</td>
<td>20001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Cell Configuration</td>
<td>20102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Allocation to Groups</td>
<td>20103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send Individual Tasks</td>
<td>20104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive Individual Tasks</td>
<td>20305</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive Individual Eng. Data</td>
<td>20306</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queue Jobs</td>
<td>20307</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set-up Part/Material</td>
<td>20408</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preparing Tools</td>
<td>20409</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The “ID” column in the above table shows the identification number assigned to the business entities.

1 It should be noted that this appendix will not discuss the rationale for defining generic entities as this has been discussed in chapters 4 to 7.
entities. The ID numbers are formatted as A-BB-CC and comprise the domain process number (i.e. A), business process number (i.e. BB) and the enterprise activity number (i.e. CC). This method of identification is used by the prototype tools and also referred by other figures and tables in appendixes of this thesis.

A.2) Reusable Function Entities
A set of semi-generic reusable function entities were defined by the semi-generic model (§6.2). These function entities (see Table A.2) represent modelling descriptions of real world resources and are categorised into three types, including human resource, physical resource and application software.

<table>
<thead>
<tr>
<th>Function Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE1 - Ordering System</td>
</tr>
<tr>
<td>FE2 - Cell Human Supervisor</td>
</tr>
<tr>
<td>FE3 - Cell Human Operator</td>
</tr>
<tr>
<td>FE4 - Scheduler / Dispatcher Tool</td>
</tr>
<tr>
<td>FE5 - Cell Supervisor Application</td>
</tr>
<tr>
<td>FE6 - Cell Operator/Group Application</td>
</tr>
<tr>
<td>FE7 - Physical Device Drivers</td>
</tr>
<tr>
<td>FE8 - Message Dialogue</td>
</tr>
<tr>
<td>FE9 - Monitoring</td>
</tr>
</tbody>
</table>

Table A.2: Library of the Building Blocks - Reusable Function Entities

A.3) Semi-Generic Information Entities
Table A.3 (includes five tables) classifies the information entities defined in this research to support information requirements of the semi-generic model at the conceptual level (§5.3.2.2). These information entities include Enterprise Objects (EO), Object Views (OV), and Information Elements (IE).

Table A.3: Library of the Building Blocks - Reusable Information Entities (including 5 tables)

<table>
<thead>
<tr>
<th>EO-01 Ordering management</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE-011 order date</td>
</tr>
<tr>
<td>IE-012 batch no.</td>
</tr>
<tr>
<td>IE-013 part identifier</td>
</tr>
<tr>
<td>IE-014 order description</td>
</tr>
<tr>
<td>IE-015 total quantity</td>
</tr>
<tr>
<td>IE-016 due date</td>
</tr>
<tr>
<td>IE-017 owner</td>
</tr>
</tbody>
</table>

Table A.3.1
### EO-02 Task management

<table>
<thead>
<tr>
<th>IE-021</th>
<th>task identifier</th>
<th>OV-21</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE-022</td>
<td>task description</td>
<td></td>
</tr>
<tr>
<td>IE-023</td>
<td>quantity</td>
<td></td>
</tr>
<tr>
<td>IE-024</td>
<td>place identifier</td>
<td></td>
</tr>
<tr>
<td>IE-025</td>
<td>operator identifier</td>
<td></td>
</tr>
<tr>
<td>IE-026</td>
<td>device identifier</td>
<td></td>
</tr>
<tr>
<td>IE-027</td>
<td>priority</td>
<td></td>
</tr>
<tr>
<td>IE-028</td>
<td>start time</td>
<td></td>
</tr>
<tr>
<td>IE-029</td>
<td>finish time</td>
<td></td>
</tr>
<tr>
<td>IE-0210</td>
<td>no. of operations</td>
<td></td>
</tr>
<tr>
<td>IE-0211</td>
<td>transporting task identifier</td>
<td></td>
</tr>
<tr>
<td>IE-0212</td>
<td>transporting device identifier</td>
<td></td>
</tr>
<tr>
<td>IE-0213</td>
<td>transporting operator identifier</td>
<td></td>
</tr>
<tr>
<td>IE-0214</td>
<td>transporting start time</td>
<td></td>
</tr>
<tr>
<td>IE-0215</td>
<td>transporting finish time</td>
<td></td>
</tr>
<tr>
<td>IE-0216</td>
<td>transporting item</td>
<td></td>
</tr>
<tr>
<td>IE-0217</td>
<td>material identifier</td>
<td></td>
</tr>
<tr>
<td>IE-0218</td>
<td>material amount</td>
<td></td>
</tr>
<tr>
<td>IE-0219</td>
<td>initial place of material</td>
<td></td>
</tr>
<tr>
<td>IE-0220</td>
<td>destination place of material</td>
<td></td>
</tr>
<tr>
<td>IE-0221</td>
<td>material description</td>
<td></td>
</tr>
<tr>
<td>IE-0222</td>
<td>tools kit identifier</td>
<td></td>
</tr>
<tr>
<td>IE-0223</td>
<td>tools amount</td>
<td></td>
</tr>
<tr>
<td>IE-0224</td>
<td>initial place of tools</td>
<td></td>
</tr>
<tr>
<td>IE-0225</td>
<td>destination place of tools</td>
<td></td>
</tr>
<tr>
<td>IE-0226</td>
<td>tools description</td>
<td></td>
</tr>
<tr>
<td>IE-0227</td>
<td>auxiliary equipment identifier</td>
<td></td>
</tr>
<tr>
<td>IE-0228</td>
<td>initial place of equipment</td>
<td></td>
</tr>
<tr>
<td>IE-0229</td>
<td>destination place of equipment</td>
<td></td>
</tr>
<tr>
<td>IE-0230</td>
<td>equipment description</td>
<td></td>
</tr>
</tbody>
</table>

#### Task List

- IE-0227

#### Material List

- IE-0218

#### Tool Kit

- IE-0223

#### Auxiliary Equipment

- IE-0227

---

### EO-04 Monitoring

<table>
<thead>
<tr>
<th>IE-041</th>
<th>time/data</th>
<th>OV-41</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE-042</td>
<td>group n status</td>
<td></td>
</tr>
<tr>
<td>IE-043</td>
<td>operator x status</td>
<td></td>
</tr>
<tr>
<td>IE-044</td>
<td>device n status</td>
<td></td>
</tr>
<tr>
<td>IE-045</td>
<td>from date</td>
<td></td>
</tr>
<tr>
<td>IE-046</td>
<td>to date</td>
<td></td>
</tr>
<tr>
<td>IE-047</td>
<td>order identifier</td>
<td></td>
</tr>
<tr>
<td>IE-048</td>
<td>task identifier</td>
<td></td>
</tr>
<tr>
<td>IE-049</td>
<td>job identifier</td>
<td></td>
</tr>
<tr>
<td>IE-0410</td>
<td>group name</td>
<td></td>
</tr>
<tr>
<td>IE-0411</td>
<td>operator name</td>
<td></td>
</tr>
<tr>
<td>IE-0412</td>
<td>device identifier</td>
<td></td>
</tr>
<tr>
<td>IE-0413</td>
<td>result (OK Scrap)</td>
<td></td>
</tr>
<tr>
<td>IE-0414</td>
<td>due date</td>
<td></td>
</tr>
<tr>
<td>IE-0415</td>
<td>delay time</td>
<td></td>
</tr>
<tr>
<td>IE-0416</td>
<td>report from date</td>
<td></td>
</tr>
<tr>
<td>IE-0417</td>
<td>report to date</td>
<td></td>
</tr>
<tr>
<td>IE-0418</td>
<td>item (options)</td>
<td></td>
</tr>
</tbody>
</table>

#### Dynamic Data

- OV-41

#### History Data

- OV-42

#### Formatted Data

- OV-44

(And OV-36 Sys. Static Data)

---

#### Report

- OV-43

(And History Data OV-42)

---

Table A.3.2

Table A.3.3
### EO-03 Supporting Data

| IE-031 | part identifier |
| IE-032 | operation identifier (job list) |
| IE-033 | operation description |
| IE-034 | operation time (min) |
| IE-035 | set-up time (min) |
| IE-036 | loading time(min) |
| IE-037 | unloading time(min) |
| IE-038 | tools requirements |
| IE-039 | auxiliary equipment requirements |
| IE-040 | device program identifier |
| IE-041 | device identifier |
| IE-042 | alternative device identifier |
| IE-043 | device program identifier |
| IE-044 | device identifier |
| IE-045 | device name |
| IE-046 | device type |
| IE-047 | device capability size |
| IE-048 | planned maintenance time |
| IE-049 | availability time |
| IE-050 | device controller identifier |
| IE-051 | not-working item |
| IE-052 | not-working from date |
| IE-053 | not-working to date |
| IE-054 | cell element identifier |
| IE-055 | cell element connected to |
| IE-056 | cell element access to |
| IE-057 | cell element on host |
| IE-058 | cell element user identifier |

### OE-05 Messaging

| JE-051 | message string |
| JE-052 | message type identifier (options) |
| JE-053 | message receiver identifier |
| JE-054 | message sender identifier |

Table A.3.4

Table A.3.5
A.4) Reference Procedural Rules

The "reference procedural rules" (§6.5.6) specify the flow of activities associated with the semi-generic model. Table A.4 illustrates the rules defined as part of this research. The rules include events, "next activities", end-status and corresponding activities that should not be performed (referred as "passes activities"), when particular end-status conditions occur. The "reference procedural rules" should be modified when developing particular models. The new rules termed as "allocated procedural rules", specify the operation flow for specific models (also see appendix B - §B.2). The textural format of these rules can be used with other tools (e.g. Petri-Nets) to operationalize models during the real time execution life phase.

<table>
<thead>
<tr>
<th>Event</th>
<th>E.A ID</th>
<th>End Status</th>
<th>Next Activity</th>
<th>Passed Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Cell AND New Order</td>
<td>10101</td>
<td>OK</td>
<td>10102</td>
<td>10100 10100 10100 10100 10100 10100 10100 10100 10100 10100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Abort 10103</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10102 OK 10103</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10103 OK 10104</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10104 OK 10114</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10114 OK 10102</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>101013 Abort Terminate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>101012 Abort Terminate</td>
</tr>
<tr>
<td>.AND. Scheduling Request</td>
<td>10505</td>
<td>Done</td>
<td>10507</td>
<td></td>
</tr>
<tr>
<td>Operator Confirmation</td>
<td>20517</td>
<td>OK</td>
<td>20514</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refused</td>
<td>Terminate</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>20514</td>
<td>Done</td>
<td>20515 20515</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20515</td>
<td>Done</td>
<td>20516</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20516</td>
<td>Message</td>
<td>20623 20623</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Next</td>
<td>20305</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>End</td>
<td>Terminate</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>20623</td>
<td>Error</td>
<td>20618 20618</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command</td>
<td>20619</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Emergency</td>
<td>20620</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Report</td>
<td>20621</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Normal</td>
<td>20622</td>
<td></td>
</tr>
<tr>
<td>Start Monitoring Process</td>
<td>30011</td>
<td>Specific</td>
<td>31201 31300 31400</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>On-line</td>
<td>31306 31307 31400</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31308</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Report</td>
<td>31409 31300 31300</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31201 Done</td>
<td>31202</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31202 Done</td>
<td>31203</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31203 Done</td>
<td>31204</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31204</td>
<td>31205 31205</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31205 Terminate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.OR. Operator Cancellation</td>
<td>31306</td>
<td>Loop</td>
<td>31308</td>
<td></td>
</tr>
<tr>
<td>.OR. Operator Cancellation</td>
<td>31307</td>
<td>Loop</td>
<td>31308</td>
<td></td>
</tr>
<tr>
<td>.OR. Operator Cancellation</td>
<td>31308</td>
<td>Loop</td>
<td>31306 31307</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31409 Done</td>
<td>Terminate</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Error</td>
<td>30011</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31410 Done</td>
<td>Terminate</td>
<td></td>
</tr>
</tbody>
</table>

Table A.4: Library of the Building Blocks – Reference Procedural Rules
A.5) Particular System Elements – The Default Configuration

As discussed in section 6.6, the cell application tools (CATools) and their elemental software components were developed based on the “system components” and “particular system elements”. The “system components” comprise generic system entities defined by the semi-generic model. The “particular system elements” include the modelling specifications that cannot be defined generically and should be specified for particular manufacturing cells and cell control systems. It was discussed in section 6.6 that to develop prototype application tools a default configuration had to be determined for these elements. However, in a practical case this configuration should be redefined based on real case data.

The default configuration assumed was as follows.

A generic cell was assumed to be divided into three groups including one supervisor group and two operator groups (supported by CATool-1 and CATool-2). One human resource was assigned to each group including one supervisor (i.e. FE2) and two operators (i.e. FE3). DP1 (Task Management), DP3 (Monitor Cell) and BP6 (Generate message) associated with DP2 were assigned to the supervisor group. DP2 (Operation Management) was allocated to operator groups with different accessibility to the “engineering data” for each operator, based on the use of CATool-2A and CATool-2B. The “allocated procedural rules” were also defined for BE’s allocated to the groups, as illustrated in Appendix B. The “connectivity functions” and “data manipulation functions” were embedded into software components by using of CIMBIOSYS and SQL (using Ingres database management software) functions codes. Examples of these codes are shown in Appendix D.

A.6) Reusable Software Components

The software components provide functional capabilities in the form of software applications that correspond to the smallest units of activities described by the semi-generic model. According to the CIMOSA modelling structure, the smallest units of activities are the enterprise activities. However, at the model execution level, CIMOSA enterprise activities may be decomposed further into several fragments of activity, referred to as function operations (FO). The function operations should be considered as being atomic units of functional capabilities. A FO is mainly engineering data, which can take the form of a fragment of a computer program (i.e. codes), or a software tool or a command message. FO’s need to be defined by system developers and affixed to related EA’s. Thus execution of the model actually corresponds to carrying out EA’s (and their elemental FO’s) in the correct sequence based on the use of procedural rules. Theoretically, one software component can be defined for each function operation, as

![Figure A.1: The level of functionality for designing the software components](image)
shown by figure A.1. However, on considering the complexity of the semi-generic model developed in this research, and the need to achieve manageable results from this research in a limited time, it was decided that software components would be developed as big as building blocks, i.e. at a grain size that corresponds to a capability to realise CIMOSA business processes. However, further decomposition of the software components could be readily achieved at a subsequent stage, thereby providing more flexible and reconfigurable modelling system especially when generating the cell application tools (CATools). It is presumed that a future use of suitable software engineering approaches and advanced technology by software vendors could lead to the commercial availability of software components at various levels granularity to suit specific domains of manufacturing systems.

Table A.5 provides descriptions of the sample set of software components required to realise the business processes defined by the semi-generic model. Later in this project, a set of software components were designed and implemented based on these descriptions and were used as building blocks of the prototype cell application tools (see appendix D).

Software components can in principle be standalone application tools, function codes, software module, applets, etc., depending on the method used to build and execute applications. Each software component has an embedded capability to use the infrastructure services available within the modelling environment.

In this study however, the software components were built based on the semi-generic business processes and cannot be changed during system operation. Should a system developer require to change the set of business processes, then new software components must also be generated. This could require a major re-engineering effort. Change capability of the modelling structure is discussed in chapter 9.

---

1 The semi-generic model defined in this research is a relatively large model including 3 domain processes, 15 business processes, and 65 enterprise activities (each enterprise activity should typically include 5 to 10 function operations).
<table>
<thead>
<tr>
<th>Software ID</th>
<th>BP</th>
<th>Associated Functions</th>
<th>Input</th>
<th>Output</th>
<th>Function</th>
<th>Data Format</th>
<th>Builders</th>
<th>Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC101</td>
<td>BP-10100</td>
<td>List of the new orders from production dept or MRP system</td>
<td>Confirmation or refusal message for new orders</td>
<td>Accessing cell databases to check capability for performing and meeting the due-dates</td>
<td>G</td>
<td>A, B</td>
<td>Starts by frequently checking the new orders – Ends by transmitting message to the cell supervisor</td>
<td></td>
</tr>
<tr>
<td>SC105</td>
<td>BP-10500</td>
<td>Command message from Cell Supervisor</td>
<td>Arrays of static and dynamic data for specific time/date</td>
<td>Collecting data by accessing cell databases for static data and physical devices (e.g. sensors, drivers) for dynamic data</td>
<td>G, C, B</td>
<td></td>
<td>Triggers by scheduler – Ends by collecting last piece of data (or error messages)</td>
<td></td>
</tr>
<tr>
<td>SC106</td>
<td>BP-10600</td>
<td>Cell Data</td>
<td>Scheduled Lists</td>
<td>Scheduling tasks, tools, material, equipment processes</td>
<td>D, E, F</td>
<td></td>
<td>Starts by cell supervisor – End by terminating the processes</td>
<td></td>
</tr>
<tr>
<td>SC104</td>
<td>BP-10400</td>
<td>Cell Data – Scheduled Lists</td>
<td>Task lists</td>
<td>Dispatching the scheduled lists to the groups/operators</td>
<td>D, E, F</td>
<td></td>
<td>Starts by cell supervisor – End by terminating the processes</td>
<td></td>
</tr>
<tr>
<td>SC201</td>
<td>BP-20100</td>
<td>List of new orders</td>
<td>Individual tasks</td>
<td>Distribution of specific tasks</td>
<td>G</td>
<td>A, B</td>
<td>Starts automatically – Ends by terminating the processes</td>
<td></td>
</tr>
<tr>
<td>SC203</td>
<td>BP-20300</td>
<td>Specific tasks</td>
<td>Jobs list</td>
<td>Preparing required data for performing a task</td>
<td>D</td>
<td>A, B</td>
<td>Starts by frequently checking the new orders – Ends by transmitting message to the cell groups</td>
<td></td>
</tr>
<tr>
<td>SC204</td>
<td>BP-20400</td>
<td>Jobs list</td>
<td>List of operator activities</td>
<td>Receive the job list and arrange activities to prepare the cell resources for the jobs</td>
<td>G</td>
<td>A, B</td>
<td>Starts automatically – Ends by terminating the processes</td>
<td></td>
</tr>
<tr>
<td>SC205</td>
<td>BP-20500</td>
<td>Individual Job</td>
<td>Execution messages to the physical devices, keeping the records of the jobs</td>
<td>After operator confirmation, job specifications should be passed to the physical execution system via the device drivers</td>
<td>D, A, B, G</td>
<td>F</td>
<td>Starts with operator confirmation – Ends by terminating the processes</td>
<td></td>
</tr>
<tr>
<td>SC206</td>
<td>BP-20600</td>
<td>-</td>
<td>Messages</td>
<td>Generate and transmit messages</td>
<td>G</td>
<td>A, B</td>
<td>Starts with external event (e.g. operator decision) – Ends with terminating transmission and receiving confirmation</td>
<td></td>
</tr>
<tr>
<td>SC312</td>
<td>BP-31200</td>
<td>User selection</td>
<td>Formatted data</td>
<td>Receive request for specific data set, collect data, format and display it</td>
<td>G</td>
<td>A, B</td>
<td>Starts with operator request – Ends with displaying the data or transmitting an error message</td>
<td></td>
</tr>
<tr>
<td>SC313</td>
<td>BP-31300</td>
<td>Command message</td>
<td>Specific data set</td>
<td>Repetitively collect data and display in a specified frequency</td>
<td>G</td>
<td>A, B</td>
<td>Starts with command messages – Continuously operates until interrupts by an external event</td>
<td></td>
</tr>
<tr>
<td>SC314</td>
<td>BP-31400</td>
<td>Monitoring data set</td>
<td>Cell status report</td>
<td>Receive and analyse monitoring data and generate reports</td>
<td>G</td>
<td>A, B</td>
<td>Starts with command messages – Ends with terminating processes</td>
<td></td>
</tr>
</tbody>
</table>

A = Commonly used GUI builders such as Visual Basic, Java, C++, SUIT, ZAP, GUPTA, etc.
B = Appropriate functions should be embedded into the software (e.g. SQL and CIMBIOSYS functions) to enable interacting with the infrastructure services
C = A specifically designed tool for particular cell support tool, e.g. interface to connect the cell data and deliver to the cell scheduler,
D = The data format depends on the specific supporting tool which should be changed to the unified format used within the infrastructure system,
E = Commercial supporting tools e.g. scheduler, DBMS, etc.
F = Interface to the external elements e.g. physical devices, supporting tools,
G = Structured text or binary data

Table A.5: Library of the Building Blocks – Software Components Descriptions
## Appendix B

Sample Templates and Procedural Rules

This appendix includes examples of templates generated for the semi-generic model by using the SEWOSA CASE tool. It also describes an example Reference Procedural Rules developed during this research and the way that Allocated Procedural Rules should be generated based on the allocation of enterprise activities to each cell group.

### DOMAIN

- **Part 1:** Domain Description
  - **Type:** <select from list>
  - **Identifier:** DM-1
  - **Name:** Manufacturing Cell System
  - **Design Authority:** enrp
  - **Description:** Manufacturing Cell Control Model
  - **CIMOSA Compliant:** yes

- **Part 2:** Domain Components
  - **Domain Objectives:** carry out cell tasks, minimize process time, flexibility, run-time supervision on cell elements, monitor orders condition
  - **Domain Constraints:** resource availability, capability for specific task engineering support data availability
  - **Domain Processes:** New Tasks management DP1, Operation Management DP2, Monitoring Cell DP3

- **Boundary:** Cell Requirements, Activity Procedures, Activity Procedures Consumption

- **Object Views:** order information, raw Mat Parts, Finished Parts, cell status info

- **Events:** Start cell process, Finish cell process

### BUSINESS PROCESS

- **Type:** <select from list>
- **Identifier:** BP-2
- **Name:** Operate specific task
- **Design Authority:** <authorised person>
- **Description:**
  - **Objective:** carry out cell tasks, start a task, pass it to related group and finish a task, send end process messages, send emergency messages, send error messages
  - **Constraint:** resource availability, resource capability, meet required quality, meet required quantity
  - **Declarative Rule:**
  - **Function Input:** Eng. data, system static data, sys configuration data, raw Mat Parts, program(tape disk), equipment, tools, new work to list
  - **Function Output:** Finished parts, messages, results history file, return materials
  - **Control Input:** new WTL, start operation
  - **Control Output:** Finish cell process
  - **Resource Input:**
  - **Resource Output:**
  - **Comprises:** Data Preparation - Resource Preparation - Operate job - generate messages
  - **Behaviour:**
  - **On (Start):** DO Data Preparation
  - **On (ES(Data Preparation) = ok):** DO Resource Preparation
  - **On (ES(Resource Preparation) = ok):** DO Operate a job
  - **On (ES(operate a job) = end):** DO FINISH
  - **On (ES(operate a job) = message):** DO generate messages
  - **On (ES(generate messages) = return):** DO Operate a job
  - **On (ES(generate messages) = end):** DO FINISH
  - **On (ES(operate a job) = next):** DO Data Preparation

---

![Figure B.1: Examples of SEWOSA Templates](image-url)
### Reference Procedural Rules

<table>
<thead>
<tr>
<th>Events</th>
<th>EA</th>
<th>Endf</th>
<th>Next</th>
<th>Passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>New_Order</td>
<td>10101</td>
<td>OK</td>
<td>10102</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10101</td>
<td>ABORT</td>
<td>10103</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10102</td>
<td>OK</td>
<td>10103</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10102</td>
<td>ABORT</td>
<td>10103</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10103</td>
<td>OK</td>
<td>10104</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10103</td>
<td>ABORT</td>
<td>10104</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10104</td>
<td>OK</td>
<td>10103</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10104</td>
<td>ABORT</td>
<td>10103</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10114</td>
<td>OK</td>
<td>10113</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10114</td>
<td>ABORT</td>
<td>10113</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10012</td>
<td>ANY</td>
<td>TERMINATE</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td>Schedule</td>
<td>10505</td>
<td>DONE</td>
<td>10507</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10507</td>
<td>ANY</td>
<td>10606</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10606</td>
<td>DONE</td>
<td>10608</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10608</td>
<td>DONE</td>
<td>10609</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10609</td>
<td>DONE</td>
<td>10610</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>10610</td>
<td>DONE</td>
<td>TERMINATE</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td>Dispatch</td>
<td>20101</td>
<td>DONE</td>
<td>20102</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>20102</td>
<td>DONE</td>
<td>20103</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>20103</td>
<td>DONE</td>
<td>20104</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>20104</td>
<td>DONE</td>
<td>TERMINATE</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td>New_Task</td>
<td>20305</td>
<td>DONE</td>
<td>20306</td>
<td>NONE NONE NONE</td>
</tr>
<tr>
<td></td>
<td>20306</td>
<td>DONE</td>
<td>20307</td>
<td>NONE NONE NONE</td>
</tr>
</tbody>
</table>

**Note:**
The Events column defines the external events buttons in CATools.
The Next column defines the next process after current one (Max 3 simultaneously processes)
The TERMINATE statement stops the process and waits for triggering an external event.
The Ignore column defines the EA's which must be passed even if the current EA was not in EA's availability list. For example EA10111 is not in EA's availability list, therefore all EA's in Domain Process 2 (20000) must be passed.

### EA's availability

<table>
<thead>
<tr>
<th>Events</th>
<th>EA</th>
<th>Passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>New_Order</td>
<td>10101</td>
<td>10102</td>
</tr>
<tr>
<td></td>
<td>10102</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10102</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>10103</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10103</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>10104</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10104</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>10114</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10114</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>10013</td>
<td>TERMINATE</td>
</tr>
<tr>
<td></td>
<td>10012</td>
<td>TERMINATE</td>
</tr>
<tr>
<td>Schedule</td>
<td>10505</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10506</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10507</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10606</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10607</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10608</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10609</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10610</td>
<td>DONE</td>
</tr>
<tr>
<td>New_Task</td>
<td>20305</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>20306</td>
<td>DONE</td>
</tr>
</tbody>
</table>

### Allocated Procedural Rules

<table>
<thead>
<tr>
<th>Events</th>
<th>EA</th>
<th>Passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>New_Order</td>
<td>10101</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10102</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10102</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>10103</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10103</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>10104</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10104</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>10114</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>10114</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>10013</td>
<td>TERMINATE</td>
</tr>
<tr>
<td></td>
<td>10012</td>
<td>TERMINATE</td>
</tr>
<tr>
<td>Schedule</td>
<td>10505</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10506</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10507</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10606</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10607</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10608</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10609</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>10610</td>
<td>DONE</td>
</tr>
<tr>
<td>New_Task</td>
<td>20305</td>
<td>DONE</td>
</tr>
<tr>
<td></td>
<td>20306</td>
<td>DONE</td>
</tr>
</tbody>
</table>
Appendix

EXPRESS Representation of the Information Model

This appendix illustrates part of the semi-generic information model developed using the EXPRESS language (§6.3.2) and represented as EXPRESS-G diagrams. In addition, example EXPRESS codes associated with the diagrams are included.

Figure C.1: Graphical illustrations of the EXPRESS model developed for the semi-generic information model (including 8 diagrams)

**Complete Schema Level Diagram for Manufacturing Cell Data**

**Figure C.1.1**

**Complete Entity Level Diagram for Messaging System**

**Figure C.1.2**
Complete Entity Level Diagram for Supporting Data

Eng_Data

Configuration_Data

Sys_Static_Data

Persons_Data

Device_data

Temp_Not_working

Complete Entity Level Diagram for Monitoring System

Dynamic_Data

History_Data

Formatted_Data

Report

Supporting_Data_schema.Sys_Static_Data

Figure C.1.3

Figure C.1.4

183
Complete Entity Level Diagram for Task Management

Figure C.1.5
Complete Entity Level Diagram for Cell Controller Schema

Figure C.1.6
Complete Entity Level Diagram for Task Management Schema

- **Manufacturing_Cell_Data**
  - **Ordering**
    - **New_Order_List**
      - `order_date`
      - `task_no`
      - `person_id`
      - `due_date`
      - `owner`
      - `total_amount`
  - **Message_dialouge**
    - `task_id`
    - `msg_type_id`
    - `msg_sender_id`
    - `msg_subject`
    - `msg_content`
    - `msg_status`
    - `msg_time`
  - **Task_Management**
    - **Main_Task**
      - `query`
      - `query_id`
      - `query_name`
      - `query_type`
      - `query_goal`
      - `query_time`
      - `query_status`
      - `query_priority`
    - **Scheduling_rule**
      - `rule_id`
      - `rule_name`
      - `rule_type`
      - `rule_effect`
      - `rule_time`
      - `rule_status`
      - `rule_priority`
    - **Transporting_Task**
      - `task_id`
      - `task_type_id`
      - `task_status_id`
      - `task_due_date`
      - `task_owner`
      - `task_source`
      - `task_destination`
      - `task_comment`
      - `task_time`
    - **Material**
      - `material_id`
      - `material_name`
      - `material_type`
      - `material_quantity`
      - `material_cost`
    - **Tools**
      - `tool_id`
      - `tool_name`
      - `tool_type`
      - `tool_quantity`
      - `tool_cost`
    - **Auxiliary_Equipment**
      - `equipment_id`
      - `equipment_name`
      - `equipment_type`
      - `equipment_quantity`
      - `equipment_cost`
  - **Supporting_Data**
    - **Eng_Data**
      - `eng_id`
      - `eng_name`
      - `eng_type`
      - `eng_effect`
      - `eng_time`
      - `eng_status`
      - `eng_priority`
    - **Configuration_Data**
      - `conf_id`
      - `conf_name`
      - `conf_type`
      - `conf_effect`
      - `conf_time`
      - `conf_status`
      - `conf_priority`
    - **Sys_Static_Data**
      - `sys_id`
      - `sys_name`
      - `sys_type`
      - `sys_effect`
      - `sys_time`
      - `sys_status`
      - `sys_priority`
    - **Persons_Data**
      - `person_id`
      - `person_name`
      - `person_type`
      - `person_effect`
      - `person_time`
      - `person_status`
      - `person_priority`
    - **Device_data**
      - `device_id`
      - `device_name`
      - `device_type`
      - `device_effect`
      - `device_time`
      - `device_status`
      - `device_priority`
    - **Temp_Not_working**
      - `temp_id`
      - `temp_name`
      - `temp_type`
      - `temp_effect`
      - `temp_time`
      - `temp_status`
      - `temp_priority`
  - **Dynamic_Data**
    - `dynamic_id`
    - `dynamic_name`
    - `dynamic_type`
    - `dynamic_effect`
    - `dynamic_time`
    - `dynamic_status`
    - `dynamic_priority`
  - **History_Data**
    - `history_id`
    - `history_name`
    - `history_type`
    - `history_effect`
    - `history_time`
    - `history_status`
    - `history_priority`
  - **Formatted_Data**
    - `formatted_id`
    - `formatted_name`
    - `formatted_type`
    - `formatted_effect`
    - `formatted_time`
    - `formatted_status`
    - `formatted_priority`
  - **Report**
    - `report_id`
    - `report_name`
    - `report_type`
    - `report_effect`
    - `report_time`
    - `report_status`
    - `report_priority`
Sample EXPRESS Codes (integrity checked by ICE software [154])

(* NO error, Checked by ICE, 24 July 97*)

SCHEMA cell_controller_Schema ;
USE FROM Date_Time_Schema (Date_Time);
USE FROM Data_Base_Schema (Operations);
USE FROM Modification_Schema (Modification);
USE FROM Device_Controller_Schema (Device Controller);
USE FROM Schedule_Dispatch_Schema (Work_To_List);

ENTITY Task
SUPER TYPE OF (Dist_Task);
task_id : STRING;
task_name : OPTIONAL STRING;
batch_no : STRING;
quantity : INTEGER;
priority : INTEGER;
task_desc : OPTIONAL STRING;
due_date : Date_Time;
END ENTITY;

ENTITY Dist_Task
SUBTYPE OF (Task);
g_name : STRING;
sub_wtl : WORK_TO_LIST (*WHERE g_name = group_name ;*)
END_ENTITY;

ENTITY Supervisor
SUPERTYPE OF (Group); (* Possibly ABS*)
new_task : Task;
io_mess : LIST [0:?] OF Message;
group_task : LIST [0:?] OF Dist_Task;
configuration : Modification;
display : LIST [0:?] OF Monitoring;
de : Device Controller;
human_supervisor_job : Human_Super_Task

operator_status : OPTIONAL
dev_gro_oper_status ;
END_ENTITY;

TYPE task_job_status = ENUMERATION OF (ACTIVE, DONE);
END_TYPE;

ENTITY Message
sender : STRING;
receiver : STRING;
content : STRING;
mess_type : message_type
END_ENTITY;

TYPE message_type = ENUMERATION OF (COMMAND, ERROR, EMERGENCY, CONFIRM, NORMAL);
END_TYPE;

ENTITY Auto_Manual_Load_Unload

at_time : Date_Time;
on_device_id : Dist_Task;
operator_name Dist_Task;
(*could be an automated device*)
load_part_name : OPTIONAL Operations;
load_tool_name : OPTIONAL Operations;
load_equipment_name : OPTIONAL Operations;
load_program_name : OPTIONAL Operations;

END_ENTITY;

ENTITY Human_Super_Task
END ENTITY;

ENTITY Analyse_Report
END_ENTITY;

END_SCHEMA;

SCHEMA Schedule_Dispatch_Schema ;
USE FROM cell_controller_Schema (Monitoring, Task);
USE FROM Date_Time_Schema (Date_Time);
USE FROM Data_Base_Schema ;
ENTITY Schedule_Dispatch
SUPERTYPE OF (Schedule);
END_ENTITY;
ENTITY Schedule
SUPERTYPE OF (Dispatch);

188
SUBTYPE OF (Schedule Dispatch);
dyn_data : Dynamic_Data;
stat_data: Static_Data;
prepare_schedule : Schedule_List;
END ENTITY;

ENTITY Dispatch
SUBTYPE OF (Schedule);
prep_schedule : Schedule_List;
wtl: Work_To_List;
exact time : Date_Time;
config_data : Configuration;
person_data : Person;
setup_data : Setting_Up
break down : Temporary_Not_Working ;
END ENTITY;

ENTITY Work_To_List;
task id : STRING;
task name : OPTIONAL STRING;
task_duedate : Date_Time;
start loading time : Date_Time;
start_unloading time : Date_Time;
start_working_time : Date_Time;
finish_working_time: Date_Time;
quantity :Task;
group_name :Device_List;
(*others are found based on id *)
device id: Device List
operator_name :Device List;
priority in group : INTEGER;
END ENTITY;

Entity Dynamic_Data;
capture_dynamic_data: Monitoring;
END_ENTITY;

Entity Static_Data;
task_data : Task;
opration_data: Operations;
device_ data: Device_List;
duedate: Date_Time;
END ENTITY;

ENTITY Schedule_List ;
END ENTITY;
END_SCHEMA;

SCHEMA Data_Base_Schema;

USE FROM cell_controller_Schema
(Message,Task,Analyse_Report,Dist Task);
USE FROM Schedule_Dispatch_Schema;
USE FROM DateTime_Schema (DateTime);

Entity Cell_Tables
SUPERTYPE OF (ONEOF
(Operations, Device_List, Person,
CONFIGURATION, Message_History,
Temporary_Not_Working, Setting_Up, Results));
END ENTITY;

ENTITY Operations
SUBTYPE OF (Cell_Tables);
task id : STRING;
task operation : STRING;
device id : STRING;
setup_time : Setting_UP;
equipment: STRING;
tools: STRING;
program no: STRING;
description: OPTIONAL STRING;
working_time mm : INTEGER;

END ENTITY;

ENTITY Device_List
SUBTYPE OF (Cell_Tables);
device name : STRING;
device id : STRING;
controller_id : STRING ;
group name : STRING;
dev_working_time : INTEGER;
planned_maint time : OPTIONAL INTEGER ;
END ENTITY;

ENTITY Person
SUBTYPE OF (Cell_Tables);
name : STRING;
position : STRING;
place : STRING;
skill_level : STRING ;
working_time :work_time;

END ENTITY;

ENTITY Temporary_Not Working
SUBTYPE OF (Cell_Tables);
item id: STRING;
from_date: Date_Time;
to_date : Date_Time;

END_ENTITY;

ENTITY Setting Up
SUBTYPE OF (Cell_Tables);
Operation id :STRING;
device id :STRING;
setuup_time: INTEGER;
load_part_time : INTEGER;
load_tools_time : INTEGER ;
load_equip_time : INTEGER;
load_program time : INTEGER;
unload_part time : INTEGER;
unload_tools_time: INTEGER;
unload_equip_time : INTEGER;
unload_program_time : INTEGER ;

END__ENTITY;

ENTITY Results
SUBTYPE OF (Cell_Tables);

END ENTITY;

END_SCHEMA;

SCHEMA Device_Controller_Schema

USE FROM Data_Base_Schema (device List);
USE FROM cell_controller_Schema (Message);

ENTITY Device_Controller;
device id LIST [1:7] OF device_List;
do_mess : LIST [0:7] OF Message;

ENDENTITY;
END_SCHEMA ;
Appendix

Prototype Tools

D.0) Introduction

A number of computer tools have been used during this research. These tools include tools developed particularly for this research, and tools developed by other research projects and commercial sectors and adopted by this research. This appendix discusses the adopted and developed tools used at different stages of this research.

As mentioned earlier a set of application tools were initially developed as part of this research to provide a lab-based cell application toolset capable of semi-automating a particular test cell (here referred as non-modelling approach – also see §4.0.1). However this initial toolset was rebuilt during the specification and development of a top-down approach to generically define cell control systems.

Table D.1 shows various prototype tools developed and used in different stages of the research.

<table>
<thead>
<tr>
<th>Modelling Systems</th>
<th>Function View</th>
<th>Information View</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Semi-Generic Level</em></td>
<td>Conceptual Design</td>
<td>SEWOSA CASE Tool</td>
</tr>
<tr>
<td></td>
<td>Detailed Design</td>
<td>Semi-Generic Modelling Tools – CIMBIOSYS-DB</td>
</tr>
<tr>
<td></td>
<td>Req. Def. Level</td>
<td>CDTool</td>
</tr>
<tr>
<td></td>
<td>Design Spec. Level</td>
<td>SEWOSA</td>
</tr>
<tr>
<td></td>
<td>Impl. Level</td>
<td>CDTool</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUIT – CIMBIOSYS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Impl. Level</td>
</tr>
<tr>
<td><em>Particular Level</em></td>
<td>CATools (incl. Software components) – CIMBIOSYS – DBMS – Device Drivers</td>
<td></td>
</tr>
<tr>
<td><em>Model Execution Level</em></td>
<td>SUIT – Ingres DB – CIMBIOSYS</td>
<td></td>
</tr>
<tr>
<td><em>Non-Modelling Approach</em></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table D.1: Computer tools implemented in this research

Firstly, the "Semi-generic Modelling Tool" (§7.4.1) was used to structure the use of model abstractions specified by the semi-generic model and to accommodate their representation and maintain by system data storage mechanism and services (see DB1 in Figure 7.3). Subsequently, the CDTool (§7.4.2) was developed to implement the modelling specification at the “conceptual design” level, and construct the “detailed design” level of the model using a component-based approach. This tool was also used to particularise the generic model for specific cells. Finally CDTool develops CATools that themselves comprise a set of software components. In addition, a set of software drivers were developed to emulate the functionality of physical devices in a typical
cell (e.g. machine, robot, and conveyor). In a real case, device drivers should be prepared individually to translate the unified data transaction format used within the modelling system (i.e. CIMBIOSYS format) to the format required for each specific physical devices.

In conformance with the information view of the semi-generic model, information objects were defined at the conceptual level (§5.3.2, §A.3) and stored in a local data storage system (i.e. DB1 of Figure 7.3). Subsequently at the detailed design level of modelling, the EXPRESS modelling language was implemented to code the information objects in a computer-processable form (§6.3.2).

D.1) SEWOSA CASE Tool
The SEWOSA tool [117] was developed by other researchers in the MSI Research Institute at Loughborough University as a CASE tool to implement and demonstrate application of the CIMOSA modelling structure. In this research SEWOSA was used to graphically represent the functional view of the semi-generic model at the conceptual level of modelling (see appendixes E and F for further discussions).

D.2) Information Modelling Tools
The EXPRESS language was selected to model supporting information required for modelling systems (§6.3). A graphical representation tool based on the use of the EXPRESS language was previously developed in MSI Research Institute using the IPSYS Meta-CASE Tool operating on SUN Workstation platform. This tool generates EXPRESS-G diagrams and their corresponding textural codes of the EXPRESS model. This model was checked and confirmed using the ICE tool [154] (a public domain software to check integrity of EXPRESS codes). In addition to this tool, SQL Parser [141] and STEP Parser [140] tools (both developed by researchers in MSI) were used respectively to: create database tables based on the EXPRESS model; and to populate the tables using STEP physical files [100]. Initially, an Ingres database management system [159] was implemented to provide data storage system for the modelling structure.

The information objects (i.e. "core data" - §7.5.2) defined at the requirement definition level of the particular models were transferred into the design specification level of the models and re-constructed in EXPRESS format. The information objects should be stored in the databases (indicated by DB2 in figure 7.3) and populated by real case data. Accordingly associated "accessing data" (§7.5.2) was attached to information objects and used during the model execution phase. Data transactions between these tools was supported by the use of CIMBIOSYS infrastructure system (this will be discussed later in this appendix).

D.3) SUIT Graphical User Interface Builder
Simple User Interface Tool (SUIT) [160] is a public domain software package designed to develop
graphical application tools in a simplified form of the C programming language. This tool mainly operates under the UNIX operating system and develops MOTIF applications. However it can also be run on PC's under Windows environment with the same 'look and feel' as its UNIX version.

The SUIT software package was widely used during research to build the user interfaces required for prototype tools. The main rationale for selecting this tool was its multi-platform capability and its being a public domain tool (free for educational purposes). However when developing component-based approach, it was found that this tool could underpin the use of a modular structure as required by software component techniques. Nevertheless, there are many other commercial user interface builders that could have been selected and used to develop the prototype tools.

D.4) Tools Developed as Part of the Bottom-Up Approach

As discussed in chapter 4 (§4.0.1), this research was initiated by investigating ways of semi-automation of an industrial cell and this led to the development of a set of application tools. The application tools were developed for a particular cell requiring two types of user interface, namely a cell supervisor tool and a cell operator tool, as shown in Figures D.1 and D.2. These tools were run over a local area network and communicate with the supporting data storage system and each other via the use of CIMBIOSYS application tools and a set of device drivers emulating physical resources.

![Figure D.1: Group application tool](image)

1 This operation is being performed manually at the present.
However, when developing the semi-generic modelling approach, these tools required further development and were upgraded to a new version which supports implementation of component-based approach, as outlined below.

**D.5) Interface to SEWOSA**

As discussed before (§7.4.1) the prototype toolset developed in this research include the “semi-generic modelling tools” that facilitate the development of semi-generic models. As one part of these tools, the SEWOSA CASE tool was used to generate CIMOSA-conformant models in a graphical form. Here a tool was required to communicate with SEWOSA and translate graphical (and textural) models into a computer format that can be used by other application tools used as part of the prototype toolset.

This tool was developed as a simple C coded application with embedded CIMBIOSYS functions that provides the required interface between the textural format of the semi-generic models at the conceptual level and the database management system. For example, the interface tool reads SEWOSA outputs files and generates SQL commands to create database tables in the data storage system.

**D.6) The CDTool and CATools**

The concept and development of the “Cell Design Tool” and the “Cell Application Tools” were discussed earlier in chapter 7. Here some technical issues involved in developing these prototype tools are discussed.

The CDTool (§7.4.2) was developed using SUIT application builder to interact with the semi-generic model and construct cell elements and cell groups for particular cells. It also configures the modelling elements and relationships between inter-cell elements, outer-cell systems and the infrastructure services to re-align the cell software capabilities with the new status of the cell, which occurs as a result of system changes (e.g. changes in cell resources and cell tasks). Technically, the re-alignment of cell software capabilities can be supported by regenerating the
Figure D.3: Initial interface of the Prototype CDTool developed in this research

"basic functionality" of the CATools, re-allocating an appropriate set of software component and re-configuring the infrastructure services. Here the basic functionality of the CATools is assumed to be part of the tool that provides connectivity within the tool, and between the tool and other system elements.

In general when using the CDTool a system designer specifies cell groups and allocates BE's, FE's, and resources to the groups. The CDTool uses an appropriate set of SUIT software modules and develops a particular CATool that is suitable for the specific cell group. Figure D.3 illustrates the initial user interface of the prototype CDTool application developed for this research. As shown by this figure (and also figure 7.4), the CDTool is divided into three major parts, including: a) the "Entity Availability" part that checks and imports predefined functions and business entities located in the library of semi-generic entities; b) the "Entity Allocation" part that assigns system entities to cell groups and defines their relationships (by this stage the particular model has been generated); and c) the software generation and system reconfiguration part, that actually builds the CATools and reorganises the use of infrastructure services.

When modelling entities were selected and retrieved from the library, the system designer defines cell groups and assigns entities required for each group. This will be carried out by highlighting particular items in the "available entities list" (see figure D.3), setting "group and resource check
boxes" and adding resources to the "allocated entity list" for each group. This process continues to assign all required entities to the group.

The next step is to establish relationships in each group between resources and EA's. Every group must be handled separately. By clicking on the "resource allocation button" (after setting the group check box item), the "resource allocation window" will be shown for a particular group. Figure D.4 illustrates an example of establishing relationships among resources in group1. In this example the group consists of one human resource, two physical devices and accesses to the database supporting tool. Also the human resource (here referred to as "John") is allocated the cell controller software end-user. The clicked check boxes show that user will be responsible for both devices as well as database interactions.

After resources were allocated, the designer should specify who is responsible for performing each EA and which device will be used for that particular EA. Clicking on the "Enterprise Activity Allocation" button opens different window to enable establishing these relationships. Resource specification should be determined separately for each EA. It was assumed that often it will be possible for a set of EA's to be carried out by a human resource on one or more physical devices.

In addition the network configuration must also be defined. The button "Networking" brings up another window to enable the designer to define network connectivity for each entity and allocates each entity to a computer host. In cases where entities are not connected to the network, the "OFF-
In the group supervisor, Tom acts as the supervisor and has access to all supporting tools and also operates one physical device. A set of EAs will be implemented in group 1. EA-20514 will be run by the user John on both milling 1 & 2.

In group 2, Mike is the application tool user and operates the Boring m/c and also has access to the database. This window shows a list of items in the cell which must be connected to the computer network or might be working Off-line. In this example the Milling 1 connected to the host Sharon.

This window shows a list of items in the cell which must be connected to the computer network or might be working Off-line. In this example the Milling 1 connected to the host Sharon.

Figure D.5: Different examples of specifying the entity relationships.
and relationships established, the "Configuration window" enables the designer to review the final cell element configurations allocated to each group as shown by figure D.6.

Having completed this stage, the executable cell application tools (CATools) should be generated for each cell group by writing and compiling predefined codes. The process of generating the CATools comprises: building the basic module of each CATool (common for each group) including the CIMBIOSYS and database accesses, and network connectivity; adding and configuring allocated software components to the basic module and establishing interoperability within the tool.

It was understood that automation of such a process (i.e. reconfiguration and generation of CATools) would be extremely difficult because of the need for pre-defining large number of cell entity combinations. Therefore in this research a limited number of SUIT software modules corresponding to the definition of the software components were developed. Each software module was programmed to perform a particular task, as described by related business processes. SUIT software modules were defined so that they correspond to generic and reusable "system components".

Figure D.6: The final system entity configuration
It was concluded that when dealing with a real industrial case, it would be necessary to implement a standard (or well-designed) software architecture and software components to be able to define and operationalize more detailed level of tasks (i.e. enterprise activity level). This can potentially provide significant degree of flexibility and re-configurability to rapidly develop appropriate cell control software capability when changes occur in the system specifications.

Figure D.7 illustrates the CATools generated for the group “Supervisor”. It was assumed that there will be at least one CATool for each group in a cell. In the example shown by this figure all EA’s have been allocated to the group “Supervisor”, therefore the CATool for this group will be generated with a full set of software capabilities (i.e. all predefined SUIT modules were allocated to this group). Figure D.8 also shows the CATool for group “Group1” of the example cell, which has less functionality in comparison to that of the group “Supervisor”.

The SUIT software allows the end-user to modify screen layouts of CATools (i.e. the user interface) readily and interactively and thereby can support different user preferences. Therefore a similar CATool used within a group by different end-user may have a different “Look and Feel” and different accessibility to the data storage system based on the types of tasks allocated to the group. On considering system security issues, each user was provided with a unique password to access the system.

The CATool user interface consists of the following parts:

**External Events button:** These buttons represent external events defined as part of a particular Business Process and modelled using SEWOSA ‘behaviour diagrams’ and are triggered by the end-user. An exceptions to this is the “Start” button (only provided for the supervisor group) which loads all CATools through CIMBIOSYS functions and the “Task History” button which displays previous tasks on the screen.

**Display Windows:** By clicking on each label in the “Display Windows” part, a set of information displays on each window to inform the supervisor about the status of the cell and displays results of executed tasks. Each window was predefined as a SUIT module.

**Tasks Windows:** The “Task Panel” in this window displays the queue of tasks along with essential information for each task. By clicking on each task, the “Engineering Data” panel retrieves and displays the details of operations contained in a task and its associated data.

**Message Passing Window:** This is a component common in all CATools and provides direct communication between human resources. This facility covers normal and emergency messages and it includes “New Messages”, alarm icons, and a list of received tasks.
Device Status Window: This is used to display the status of the physical devices at runtime. The status data is collected by a monitoring component.

Processing Message Window: This window keeps the user informed about the status of tasks and the execution of EA's.

D.7) CIMBIOSYS Applications
The CIMBIOSYS integration infrastructure system has been previously discussed in the literature review (§2.5.2) and its operation is explained further in appendix G (§G.4.3).

All prototype tools developed in this research are compatible with the CIMBIOSYS tool-set and are able to use CIMBIOSYS library functions. The CIMBIOSYS application tools (version 4) were implemented to meet infrastructure integration needs. In this research, all applications resided on the same network system except the database software, which must be remotely accessed.

The following describes certain technical issues about implementation of the CIMBIOSYS system within the prototype tools.

Application Establishment - To execute the prototype tools, first CIMBIOSYS applications must be enacted on each host within which applications are to be run.

The supervisor application is established by CIMBIOSYS and the main window of the application appears on the screen. The other group applications will be run by the supervisor application by
sending command messages. These messages will actually call a CIMBIOSYS function via the original application (sender) and will be connected to other applications (receiver, i.e. group application). The supervisor maintains its established message and exchanges channels with group applications until a termination function disconnects the communication link.

Example:

Supervisor request to establish group_1 application

\[
\text{asi_connect("group_1",print_response,(void*)"Connect")}
\]

Application termination function

\[
\text{asi_term(handle*)}
\]

Sending Messages - There are different types of message passed by CIMBIOSYS. The message type identification is independent from CIMBIOSYS functions. The type of the message is an identifier string added to the message context and separated from other text by the "@" sign. The type variable automatically gets a value by selecting a type of message. The receiver application gets the value of the data carried by the CIMBIOSYS function and then distinguishes the real context of the message from types or other variables of the message.

Example:

Typical message transacted between two applications

\[
\text{asi_send_data("supervisor", SUIT_getText(CURRENT_VALE), 80, func_ptr, user_data)}
\]

- receiver-id
- message content
- message length

Message content: "<message type>@[other identifier@]<message content>"

For example:

- "ERROR@Eng. Data Not Found".
- "COMMAND@start"
- "INFO@Request_Data_Obj@main_task"

The "sender" identifier of a message (which is specified by data model) is the name of the sender application as defined by the CIMBIOSYS configuration. In supervisor and group applications two different panels have been provided to send and receive dialogue messages. The receiving message panel applies a warning method to highlight high priority messages such as emergency messages.

Data Transactions - Referring to data object views as defined at the requirement definition modelling level and by the EXPRESS model created at the design specification modelling level, the data exchanged between cell elements is structured in an object oriented manner. This includes information elements with individual attributes defined at the entity level of the information model. These objects are distinguished by an object identifier (entity name in the EXPRESS model) and contain real formatted data corresponding to information elements for each object.

The transaction of an object is made through the information system (as part of the infrastructure services) which is a combination of SQL parser, STEP parser tools, and a database management
system (DBMS) associated with a number of functions embedded into prototype tools. For instance when a cell element such as "scheduler driver" requests a "new_order" object, initially a request must be sent to the DBMS unit in the form of an information request message transmitted via the CIMBIOSYS system.

Example:

\texttt{asi\_send\_data("dbms", "INFO@Request\_Data\_Obj@New\_Order", 80, func\_ptr, user\_data)}

This message activates the DBMS to generate appropriate SQL statements for the database to retrieve relevant data blocks for the "new order" object. At this time the data block must be sent back to the scheduler interface. This procedure introduced certain technical problems. The data exchange process should be carried out by using a number of CIMBIOSYS packets (with limitations on the maximum length of a transferable string using CIMBIOSYS ²). Therefore the author applied alternative means of transferring data which is based on reading and writing data in a temporary ASCII file. The format of the file conforms to SQL statements, which enables the data retrieved as an output from the database to be directly recorded in a file (each piece of information in a separate line). Having done that a CIMBIOSYS message is sent to the scheduler driver to inform it that the temporary data file was created. Consequently data will be extracted by a simple file reading function.

Example of a message context in response to a data object request:

\texttt{"INFO@Response\_Data\_Obj@-/temp\_new\_order"}

² The CIMBIOSYS transferable data length is 1500 characters and any data item longer than this must be sent in several packets of data. The first method suggested to solve this problem was to use the File Access utility of CIMBIOSYS by means of frequently running file reading function (asi\_read\_file) to read a packet of data at a time and saving a pointer address for last reading the position in the file. The second method is the use of Database Driver tool [171]. This tool is a complete tool-set for communicating with a database by manually writing SQL statements. This tool supports use of both INGRES and ORACLE database software packages.
The following programming codes show sample CIMBIOSYS and SQL functions embedded into the CATools to enable interaction with other services, including the data storage system (based on the use of Ingres) and CIMBIOSYS infrastructure system:

```c
/** "********"'**CIMBIOSYS FUNCTIONS****************/
void *handle;
void cbs_rec_func(who, data, length, user_data)
char *who;
unsigned char *data;
int length;
void *user_data;
{
char c;
if (length == 200) { /* 200 for normal messages */
SUIT_textList list = SUIT_getTextList(rec_mess_list, LIST);
SUIT_appendToTextList(list, data);
SUIT_setTextList(rec_mess_list, LIST, list);
SUIT_checkAndProcessInput(3);
SUIT_setText(rec_mess_list, CURRENT_VALUE, data);
}
if (length == 201) { /* 201 for returning OV21(task) */
SUIT_textList list = SUIT_getTextList(tasks_list, LIST);
SUIT_appendToTextList(list, data);
SUIT_setTextList(tasks_list, LIST, list);
SUIT_checkAndProcessInput(3);
SUIT_setText(tasks_list, CURRENT_VALUE, data);
}
if (length == 301) { /* 301 for returning OV31(eng_data) */
if_name=fopen("/home2/SUIT/DA TAleng_data_disp.tmp", 'a');
do (printf("before getc In");
c = gets(data); printf("after getc In");
if(strcmp(c,'l ')) printf(f_name,"
"); else printf(f_name,"os",c);
}while(c!=NULL)
fclose(f_name);
SUIT_sendToEditor(tasks_editor, data);
printf("!osln",data);
SUIT_checkAndProcessInput(3);
}
void est_all_func(start_pulldown)
{
int w;
for (w = 0; w < 5; w++) {
if (All_Entities[w][0] != NULL) {
sprintf(gn, "%c%d'", 'g', w);
nsl_connect(gn, NULL, (void *) "Connect");
}}
void cbs_send_func()
/******************CIMBIOSYS FUNCTIONS*******************/
```
Functions via CIMBIOSYS

```c
#include "asi.h"
#include "event.h"
#include "event.h"
#include "/usr4/ingreslfilesleqsqlca.h"

extern JJSQLCA sqlca;

char quit[4] = {""};

void *handle = NULL;

void cbs_rec_func(who, data, length, user_data)
{   char tempt[20] = {""};
    char temp2[3] = {""};
    char record[200] = {""};
    char et[t5];
    char c2[t5];
    char c3[t5];
    char c4[t5];
    char c5[t5];
    char c6[t5];
    char c7[t5];
    char c8[t5];
    char c9[30];
    char et0[30];
    char ctt[30];
    int dt;
    int d2; int d3; int d4;
    strcpy(tempt, data);
    if (length == 20)
    {   #line 45 "indr.sc" /
        llsqlnit((int)&sqlca);
        llcsOpen("patt", (int)32345, (int)6654);
        llwritio((int)O, (short)O, (int)t, (int)32, (int)0, "select place_id,
          task_id, part_id, batch_id, quantity, operator_id, device_id, start_t
          ime, finish_time from task_list
          where place_id=\"");
        llputdomio((short)O, (int)t, (int)32, (int)O, tempt);
        llcsQuery("patt", (int)32345, (int)6654);
    }   #line 46 "indr.sc" /
    while (sqlca.sqlcode == 0)
    {   #line 47 "indr.sc" /
        llsqlnit((int)&sqlca);
        if (llcsRetrieve("patt", (int)32345, (int)6654) != (int)O)
        {   llcsGetio((short)O, (int)t, (int)32, (int)14, c8);
            llcsGetio((short)O, (int)t, (int)32, (int)t4, ct);
            llcsGetio((short)O, (int)t, (int)32, (int)t4, c2);
            llcsGetio((short)O, (int)t, (int)32, (int)t4, c3);
            llcsGetio((short)O, (int)1, (int)30, (int)4, &dt);
            llcsGetio((short)O, (int)t, (int)32, (int)t4, c3);
            llcsGetio((short)O, (int)t, (int)32, (int)t4, c5);
            llcsGetio((short)O, (int)t, (int)32, (int)t4, c6);
            llcsGetio((short)O, (int)t, (int)32, (int)t4, c7);
            llcsERetrieve();
        }
    }   #line 49 "indr.sc" /
    sprintf(record, "%s>>>%s\t%sl%sl%dl%s1%sl%sl%s", c1, c2, c3, d1, c3, c5, c6, c7);
    dic_connect(temp2, 201, none, "Send");
    print("20 was working in");
    }
    #line 50 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 51 "indr.sc" /
    if (length == 20)
    {   "20 => *key word for search ov-21 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 52 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 53 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 54 "indr.sc" /
    if (length == 20)
    {   "20 => *key word for search ov-21 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 55 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 56 "indr.sc" /
    if (length == 20)
    {   "20 => *key word for search ov-21 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 57 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 58 "indr.sc" /
    if (length == 20)
    {   "20 => *key word for search ov-21 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 59 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 60 "indr.sc" /
    if (length == 20)
    {   "20 => *key word for search ov-21 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 61 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 62 "indr.sc" /
    if (length == 20)
    {   "20 => *key word for search ov-21 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 63 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 64 "indr.sc" /
    if (length == 20)
    {   "20 => *key word for search ov-21 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 65 "indr.sc" /
    if (length == 30)
    {   "30 => *key word for search ov-31 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
    #line 66 "indr.sc" /
    if (length == 20)
    {   "20 => *key word for search ov-21 *"   strcpy(temp2, who, 2);
        llsqlnit((int)&sqlca);
        llcsClose("patt", (int)32345, (int)6654);
    }
```
```c
llcs0pen("pat2", (int)32346, (int)2779);
llwrite((int)1, (short *)0, (int)1, (int)32, (int)0, *select operation
_id, operation_des, operation_time_min, setup_time_min, loading_time_min,
unloading_time_min, Tools_req, ax_eq_req, prog_id, device_id, all
device_id from eng_data where part_id=");
llputdomio((short *)0, (int)1, (int)32, (int)0, temp1);
llcsQuery("pat2", (int)32346, (int)2779);

/* # line 68 *indr.sc */ /* host code */
while (sqlca.sqlcode == 0) {
    /* # line 69 *indr.sc */ /* fetch */
    llsqlnit((int)&sqlca);
    if (llcsRetrieve("pat2", (int)32346, (int)2779) != (int)0) {
        llcsGetio((short *)0, (int)1, (int)32, (int)14, &c1);
        llcsGetio((short *)0, (int)1, (int)32, (int)29, &c9);
        llcsGetio((short *)0, (int)1, (int)30, (int)4, &d1);
        llcsGetio((short *)0, (int)1, (int)30, (int)4, &d2);
        llcsGetio((short *)0, (int)1, (int)30, (int)4, &d3);
        llcsGetio((short *)0, (int)1, (int)30, (int)4, &d4);
        llcsGetio((short *)0, (int)1, (int)32, (int)29, &c10);
        llcsGetio((short *)0, (int)1, (int)32, (int)29, &c11);
        llcsGetio((short *)0, (int)1, (int)32, (int)14, &c2);
        llcsGetio((short *)0, (int)1, (int)32, (int)14, &c3);
        llcsGetio((short *)0, (int)1, (int)32, (int)14, &c4);
        llcsERetrieve();
    }
}
/* # line 70 *indr.sc */ /* host code */
sprintf(record, "%s%d%d%d%d%d%s%s", c1, c9, d1, d2, d3, d4, c10, c11, c2, c3, c4);
asi_connect(temp2, none, (void *)"Connect");
asi_send_data(temp2, record, 301, none, "Send");
printf("30 was working\n");
}
/* # line 75 *indr.sc */ /* close */
llsqlnit((int)&sqlca);
llcsClose("pat2", (int)32346, (int)2779);
}
/* # line 76 *indr.sc */ /* host code */
/* # line 77 *indr.sc */ /* disconnect */
llsqlnit((int)&sqlca);
llsqlDisconnect();
/* # line 78 *indr.sc */ /* host code */
strcpy(quit, "QUIT");

void main(int argc, char *argv[])
{
    printf("before cbs\n");
    asi_init(handle, &argc, argv);
    asi_reg_receive_data_func(cbs_rec_func, ASL_NO_DATA);
    ev_manage_start((void *)handle);
    printf("after cbs\n");
}
```

```
Appendix E

Experimental Use of the SEWOSA Approach at the Design Specification Level

E.0) Introduction
This appendix reports on an initial experiment aimed at developing the semi-generic model at the "detailed design" level using the SEWOSA methodology. It discusses pro and con related to the approach. Here it is not intended to explain the methodology in detail (for a detailed discussion about the SEWOSA methodology readers are referred to Aguire PhD thesis [117]).

E.1) The SEWOSA Workbench
The Model Driven CIM research project [118] carried out at the MSI Research Institute, at Loughborough University, brought together a number of methods, tools and infrastructure utilities in different workbenches to provide a supporting toolset for modelling system life-cycles. The SEWOSA Methodology is a prominent workbench developed and implemented within the Model-Driven CIM project. The SEWOSA methodology extended the use of CIMOSA concept by using Petri-Nets techniques, object oriented design tools and integrating services of the CIMBIOSYS infrastructure [117]. The SEWOSA CASE tool was designed to model the system life-cycle from the conceptual design phase, through detailed design (model building), enactment of the model (simulation, rapid-prototyping) onto model driven operation of the system and hence physical system implementation. This CASE tool also provides computer user interfaces by using the IPSYS [172] software application, to graphically illustrate the development and execution of CIMOSA models in form of several diagrams accompanied with their textural templates.

This research used SEWOSA extensively as a CASE tool to capture and represent the modelling systems in complying with the CIMOSA architecture. At the "conceptual design" level, the SEWOSA CASE tool was used to capture and formalise system functional requirements. Later, at the "detailed design" level, this tool was experimented to model the design specification and the implementation issues of the semi-generic model (which is being discussed in this appendix).

E.2) Implementing the SEWOSA CASE Tool
Based on the use of SEWOSA methodology, interaction between generic enterprise activity (see §5.3.1), were defined at the "detailed design" level of the semi-generic model using the ‘Object Diagram’ capability of the SEWOSA CASE tool. Based on this method, a sample set of generic function entities (i.e. ‘active resource components’) was defined that interact with each other via the use of an integrating infrastructure support system. Interactions were realised by transacting messages between function operations. The function entities were defined as part of an integrating operation environment [56]. Figure E.1 illustrates part of the ‘object diagram’ developed for the semi-generic model (for a more complete illustration see appendix F).
Integrating Operation Environment

Figure E.1: Part of the 'Object Diagram' developed for the semi-generic model using the SEWOSA CASE Tool (for complete illustration see appendix F)

Activity Behaviour Diagram
Specifying the internal operation of EA's using Petri-Net method

Entity Behaviour Diagram
Specifying the internal process of the FE's and their interactions with other system entities

Figure E.2: Overview of the graphical representation of the MCC model at the detailed design level, developed using the SEWOSA CASE tool

Having defined function entities and function operations using the 'object diagram' capability of the SEWOSA CASE tool, it was also necessary to define in precise terms, the internal requirements
of each enterprise activity and their associated function entities. In this way the behavioural activity, operational flow and the status of each element (at each stage of operation) of cell were defined using ‘Activity Behaviour Diagrams’ and ‘Entity Behaviour Diagrams’ of the SEWOSA CASE tool. Use of the ‘activity behaviour diagram’ enabled the internal operation of an enterprise activity (e.g. start and ending status according to procedural rules) to be specified based on the use of predicted-action Petri-Nets [103]. This diagram also described interactions between enterprise activities and function and information entities. The ‘entity behaviour diagram’ described the behaviour of the function entities in terms of their internal processes and interaction with the other system entities.

The logical description of system functionality, so defined, needs to be mapped onto the physical resources deployed in any given system. This mapping needs to determine the configuration of the resources with respect to other system elements in a specific cell. This mapping is defined by using the ‘Resource Diagram’ capability of the SEWOSA CASE Tool.

The networking configuration of the function entities and the resources were identified using the ‘Configuration Diagram’ of SEWOSA. This diagram shows the assignment of business entity components to the computer host in which the SEWOSA tool was running. Depending on capabilities of the software and hardware of the system, a number of hosts could be employed in the networking system.

Figure E.2 illustrates a sample of the diagrams created at the design specification modelling level and relationships amongst them, to define further elements of the semi-generic model (for complete illustrations see appendix F).

The SEWOSA tool has also been deployed at the implementation level of the semi-generic model (as part of the detailed design phase). At this level, the SEWOSA tool generates a Petri-Nets model to formalise the functionality of the system and also to simulate its functionality. Subsequently actual physical resources can replace modelled entities. Figure E.3 illustrates the execution stage of the semi-generic model at the implementation level, developed using the SEWOSA CASE tool.

E.3) Discussion
Application of the SEWOSA tool was found to be appropriate and useful during the “conceptual design” level of the semi-generic model, whereas, this tool was not found to be completely suited for the research requirements at the “detailed design” level. When using the SEWOSA method, certain limitations of the method were found with respect to a real scenario that could place practical constrains on any future development of the modelling system.

Two types of limitation were identified that include:
1) Lack of sufficient support for the development of a generic modelling structure for reusable system entities,

2) Technical complexity during model implementation and execution.

To develop a modelling construction for manufacturing cell control systems and thereby to facilitate rapid re-configuration when cell conditions change, it was concluded that the modelling domain should be extended to cover a semi-generic level. It was also determined that such modelling domain should be designed based on defining generic system entities with re-usability and re-configurability attributes.

Although, the SEWOSA CASE tool provides a logical formalisation method for modelling system entities, it was envisaged that this method was suitable for modelling parts of systems that requires static relationships to be formally defined between system entities (i.e. requirement definitions). However, at a generic level, relationships between system entities cannot naturally be solidly specified. This is because at the design specification level of modelling cell systems, a range of modelling methods, tools and techniques must be used to accommodate change in cell conditions.

Hence, it was determined that the generic model should be able to define flexible links (or 'loosely defined links' [23, 57]) between system entities to maintain generality aspect of the model, as well
as an ability to particularise the model for specific cases. However such modelling characteristics were not found to be supported by the current revision of the SEWOSA CASE tool.

With respect to technical limitations, the modelling system developed in this research had to be used to produce relatively large models that cover a particular class of manufacturing cell. It was found to be necessary to encode features of a particular physical environment in a symbolic way as part of the semi-generic model. Nonetheless, this model is an abstraction of real systems. Consequently, a real system (on average) may require more comprehensive description than a semi-generic model. For instance, a type of physical resource is encoded as one function entity, although in a particular modelling case obviously more than one resource will be used and each resource may be some specialised version of the resource type modelled.

Despite the apparent conciseness of the semi-generic model, the SEWOSA CASE tool had difficulties in handling the model especially in terms of the model execution speed due to highly parallel operational characteristic of the system. Hence doubt was raised about the capability of using SEWOSA to model system closely in real time, even prior to the allocation of resources to modelled function entities. Considering that a capable computer hardware and software have been implemented in this research1, it was concluded that execution speed problems were inherited from the complexity of the SEWOSA approach that ultimately limits the size of the model2.

Bearing in mind possible limitations of the SEWOSA method regarding modelling at a semi-generic level and technical complications described, the author decided to use SEWOSA at the requirement definition level of the semi-generic model, and seek alternative approaches at the detailed design specification level (§6.4).

---

1 The execution of the model had been examined on the Sun station computer with SunOS 5.4 (Solaris) operating system.

2 The execution of an entity was carried out by transacting messages (functional operations). It was observed that for a simple function operation, a large number of messages had to be transmitted (via the CIMBIOSYS system) to achieve a particular functionality. Again it would be mentioned that no actual resource had yet been assigned to function entities.
This appendix provides a graphical representation of the semi-generic model developed using the SEWOSA CASE tool. The diagrams\(^1\) include a part of the modelling illustrations related to the conceptual design level of the semi-generic model and illustrations generated during the primary research approach at the detailed design level (i.e. the use of SEWOSA methodology – see appendix E).

Figure F1: Graphical representation of the semi-generic model (including 34 diagrams)

\(^1\) These diagrams are the actual output of the IPSYS software package used to develop the SEWOSA CASE tool. The poor graphical quality of these diagrams is related to limitations on the output format of this software package.
Functional Diagram: New Tasks management/DY-1

Order Info

Start

Collect static data

Collect dynamic data

message

Receive new order

check cell capability

collect due data

confirm cell capability

Accept new order

Refuse new order

Schedule tasks

Schedule raw material

Schedule tools

Schedule equipment

Provide Work-To-List

new work-to-list

rev WTL
Functional Diagram: Monitor Cell/DP-3

1. Start monitoring
2. Select displaying type
3. Provide query system
4. Run-time monitoring
5. Run-time data request
6. System dynamic data
7. System static data
8. Display query results
9. Format data
10. Report preparation
11. Display/print report
12. Hard printed report

- Select displaying type
- Selected option
- Specific query
- Collect specific dynamic data
- Required dynamic data
- Collect specific static data
- Required static data
- Analysis/format data
- Formatted data
- Display query results
- Formatted report data
- Report request
- System configuration data
- Continuous collecting dynamic data
- Continuous collecting static data
- New WTL
- Report preparation
Integrating Operation Environment

Integrating Infrastructure

Information Entity
Behaviour Diagram: Capacity check

START

any

FORCED

Collect static data

EA-5

done

Go/NoGo

Collect dynamic data

EA-7

any

FORCED

FINISH

Behaviour Diagram: Generate scheduled lists

START

any

FORCED

schedule tasks

EA-6

done

Go/NoGo

schedule raw material

EA-8

done

Go/NoGo

schedule tools

EA-9

done

Go/NoGo

schedule equipments

EA-10

any

FORCED

FINISH
Behaviour Diagram: Data Preparation

START

any

FORCED

receive specific task and queue

EA-5

done

Go/NoGo

retrieve specific Eng date

EA-6

done

Go/NoGo

queue jobs

EA-7

any

FORCED

FINISH

Behaviour Diagram: Resource Preparation

START

any

FORCED

set up and material

EA-6

done

Go/NoGo

preparing tools

EA-9

done

Go/NoGo

preparing requirements

EA-10

done

Go/NoGo

load program

EA-11

done

Go/NoGo

set up machine

EA-12

done

Go/NoGo

Display operator iconic

EA-13

any

FORCED
Behaviour Diagram: operate a job

START

any

FORCED

Operator confirmation

ok

Go/NoGo

run a job

GO/NO

Done

Go/NoGo

History job task

done

Go/NoGo

dequeue job task

any

FORCED

FINISH

Behaviour Diagram: generate messages

START

any

FORCED

receive heed messages

error

Go/NoGo

errors

FORCED

FORCED

FORCED

FORCED

FINISH

command

Go/NoGo

commands

FORCED

FORCED

FORCED

FORCED

FINISH

emergency

Go/NoGo

emergencies

FORCED

FORCED

FORCED

FORCED

FINISH

import

Go/NoGo

requests

FORCED

FORCED

FORCED

FORCED

FINISH

normal

Go/NoGo

normal message

FORCED

FORCED

FORCED

FORCED

FINISH
Behaviour Diagram: Display specific enquiry

START

FORCED

provide query system

FORCED

done

collect specific static data

FORCED

done

collect specific dynamic data

FORCED

done

analyse format data

FORCED

display query results

FINISH

Behaviour Diagram: On line monitoring

START

FORCED

run time monitoring

FORCED

continuously collecting static data

FORCED

continuously collecting dynamic data

Behaviour Diagram: Analysis data Generate report

START

FORCED

Report preparation

FORCED

display print report

FINISH
EA-23: Behaviour Diagram: receive send messages

EA-1: Behaviour Diagram:
EA-4: Behaviour Diagram:
check due date

EA-2: Behaviour Diagram:
Appendix

Supporting Literature

G.0) Introduction

This appendix provides supporting literature that explains issues discussed in chapter 2 (literature review) in greater detail. For the sake of completeness some material has been adopted from various resources (where indicated) with minor modification.

G.1) Review of Manufacturing Cell Definitions

Definitions of manufacturing cell were briefly discussed in chapter 2. Here some of the other well-established definitions are outlined:

- Martin [134]: "A manufacturing cell is a grouping of people, machine tools and processes into a specific area dedicated to the production of a family of parts or products. The machine tools are linked by common material handling devices under the control of a centralised cell controller."
- Franks [51]: "A group of manufacturing resources consisting of machines and workstations which are organised and scheduled as an entity to accept discrete parts, sub-assemblies, and material, the cell then adds value through processing to create a new identifiable product as its output. Cell may be automated, semi-automated, manually operated or a combination of all three types."
- Williams [52]: "Practical building blocks of CAM and CIM systems and are important as island of automation. Such cells usually consist of a number of closely co-operating different machines co-ordinated and controlled by a supervisory computer. And also "A biological analogy as the smallest autonomous unit capable of sustained production"
- Luggen [40]: "A logical arrangement of stand-alone manual or NC equipment into groups or clusters of machines to process parts by part family. Processing parts in a manufacturing cell includes completing as much of work piece processing as possible within the cell before moving it to the next sequential processing, stocking, inspection, or assembly station."
- Wommerlov [42]: "Cellular manufacturing is an application of GT (Group Technology) where a portion of a firm's manufacturing system has been converted to cells. A manufacturing cell is a cluster of dissimilar machines or processes located in close proximity and dedicated to the manufacturing of a family of parts (a cell family)."

A number of other definitions suggested by researchers have been summarised by Xiang [41]. The definition of manufacturing cells preferred in this research was discussed in chapter 2 (§2.1.1).

G.2) Modelling Definition

Modelling has been widely used to analyse various manufacturing systems [19, 117]. Modelling methods typically provide means to visualise manufacturing operations. The consensus of opinions about the definition and application of manufacturing models can be outlined as follows:

- Christensen [173]: "a structure that a system can use to simulate or anticipate the behaviour of something else"
Barkmeyer (NIST) [49]: "A model is an approximate design of some actual process or object. Engineers use models as a tool in designing and building physical objects. Similarly, system and software engineers use models to represent the characteristics of a system from several clearly points of view."

Vemadat [19]: "A model is a useful presentation of some object. It is an abstraction of reality expressed in terms of some formalism defined by modelling constructs for the purpose of the user."

Then he adds: "A manufacturing enterprise model is a consistent set of special-purpose and complimentary methods describing the various facets of an enterprise to satisfy some purpose of some business users."

Weston [57]: "A manufacturing model is a representation of some aspect of product realisation which can be used to facilitate visualisation, analysis, design, etc."

In a more technical view, an enterprise model has been defined by ANSI/NEMA [174] as:

"An enterprise model is a model of what the enterprise intends to accomplish and how it operates. It identifies the basic elements and their decomposition to any necessary degree. It also specifies the information requirements of these elements. It provides the information needed to define the requirements for integrated information systems. It is used to improve the effectiveness and efficiency of enterprise."

In some other definitions related to models [147, 175] information systems have been considered as a focal point for manufacturing modelling that supports process and resource requirements. Despite the great value of information systems in the manufacturing enterprises, the other factors such as functionality of the system, system organisation, and aspects of human elements are quite capable of being formalised as a part of an enterprise model [150, 175, 176].

In this work a manufacturing model is considered to be a set of formalised objects and methods that are constrained by some engineering assumptions made during the design of a manufacturing system. These objects and methods can be used to represent and visualise a real event in an acceptable way, provided that pre-defined assumptions are satisfied.

Models may cover the whole or part of a manufacturing enterprise. It also may support certain aspects of a system such as an information view, an activity view, and so on.

G.3) Enterprise Modelling Aspects

This section briefly discusses various aspects of manufacturing enterprise modelling as shown by figure G.1. These aspects include: purpose of modelling; things that need to be modelled; commonly used models in a manufacturing enterprise; modelling requirements; how, what and when to model; level of modelling detail; modelling decomposition; higher-order models; principle of modelling; and the benefit of modelling for manufacturing cells.
Purpose of modelling – Generally, enterprise modelling can be used by system designers to facilitate the way they analyse, control, and monitor systems. It can provide a better understanding of how an enterprise (or some part of it) works. A set of reusable entities (e.g. data, process, and knowledge) can be defined to assist rapid co-ordination of system realisation process. This can facilitate the use of advanced techniques such as simulation and rapid prototyping to anticipate events concerned with the life cycle engineering of a system [30].

What should be modelled – To increase the level of control (and efficiency) of a complex manufacturing enterprise through the modelling techniques, according to Vemadat [19] a number of essential aspects need to be modelled. These include: enterprise functionality and behaviour in terms of processes; events and activities; process of decision making; product life-cycle; physical aspects including machine, tools, equipment, layout, capacity, etc.; software applications; business data and information flow; enterprise knowledge and know-how; human factors; organisational structure e.g. level of decision making and responsibilities; exceptional events and reaction policies.

Commonly used models – Typically a model of a manufacturing enterprise may consists of various models including:

- Function model - This describes a set of sequential operations to be executed to achieve the enterprise goals [23, 61, 177].
- Product specification model - This represents detailed designed features of products as well as their geometric characteristics [38].
- Resource model - This model specifies the resources used in the system in terms of capabilities, responsibilities, physical layout, configuration, relationships, etc. [35]. Resources typically include human resource, devices and applications.
- Information model - This typically models the data required by system entities and information used to obtain or deliver data to and from the entities [67, 69]. Normally this will be accompanied with knowledge about the construction of the information system.
- Organisation model - Such model is used to describe the organisational structure of a system in terms of people who work in the system, authorities and responsibilities, their divisional location, etc. This also might be complemented by a user model, which describes human factors associated with the system.

The above issues represent various types of modelling. Although a manufacturing enterprise could also be formalised from other perspectives such as economic or finance, decision-making, inventory, customer order and so on.

Modelling Requirements - To formally represent a manufacturing system, various requirements must be met to support different perspectives on modelling [49, 57, 142]. These include:
Modelling view - This is the orientation used to determine requirements and solutions in terms of essential aspects as mentioned earlier.

Modelling methodology - The way in which the modelling process is carried out. This may be formally defined as process, guidelines, representation rules and tools used to create a model.

Modelling framework – This specifies relationships between various models in an enterprise modelling system or between various levels of abstraction within a model [178]. A framework is an interlocking set of standards or principles governing behaviour, organisation, processes, resource, communication and information among the components of a system [65, 158].

Modelling architecture – This term refers to a set of elements with clear relationships with each other. This comprises guidelines and rules for the representation of the enterprise framework, models, resources, information, labours, etc. A model architecture should point toward a meaningful organisation of an enterprise concept.

Modelling language – This provides means of formally describing the models and their relationships [178].

Modelling ontology - This is a formal description of fundamental rules of modelling, relationships, and constructs and their properties. It forms a shared terminology for the system entities (or objects) to be communicated in a given domain [179].

Modelling constructs and primitives - The atomic components of representational techniques [178].

Modelling representation: The symbology used to present the model such as graphic representation. This should meet the requirements of system users during the model implementation [55].

Modelling strategy - The primary rules for manufacturing enterprise modelling are based on providing sufficient response to the questions such as ‘why’, ‘what’, ‘how’, ‘when’ and ‘who’ to model during the life cycle of a manufacturing system [134, 180].

The ‘Why’ refers to the initial system conditions, which make a model feasible (financially or technologically). This is defined within the strategic planning phase of the manufacturing life cycle.

The ‘What’ concerns the actual operations, that are to be performed in a system and typically need to be formalised by the model at the conceptual design level of modelling. This may be addressed to functional aspects of modelling.

The ‘How’ and the ‘When’ define the way in which the formalisation will be carried out to achieve the enterprise goals (including behaviour aspects). Normally this should be specified during the
detailed design of the system. A time dimension may be attached to a static model. Under such conditions one should expect a different reaction from the same model at different instances of time. Indeed ‘When’ can also be of concern in respect of various modelling phases including: design phase, build phase, and execution phase [57].

The ‘Who’ is of concern to organisational aspects of the enterprise, and therefore specifies the people who perform the operations in the system. The ‘who’ may also be applied to people who use the model (e.g. user, developer or manager of a system).

**Modelling detail** - Various levels of detail can be chosen when modelling a product or a system life cycle. A modelling methodology can be used to specify the conceptual definition of system requirements, design phase of the life cycle, and implementation of the system model [56, 181]. Moreover, execution and maintenance phases can be partially supported by different levels of enterprise modelling detail. Although, due to many unexpected events in these phases, the model could not be as precise and reliable as other phases.

Typically the level of generality of an enterprise model can vary from a generic level to intermediate and specific level [37, 53]. At the generic level, the model can broadly define system components, rules, functionality, protocols and other fundamental issues of an unlimited area of manufacturing systems. The intermediate level of modelling targets a more specific area of industry and defines a more detailed specification of the system. At the specific level, a particular company (e.g. specific PCB manufacture) or a process in particular company will be modelled. At this level different views of models (e.g. function, information, and resources) need to be precisely formalised.

**Modelling decomposition** - When dealing with a complex system, decomposition techniques are commonly used. Breaking down the system into manageable components provides a better overall control of the system in addition to promoting possible advantage of reusability and modularity. Although, a communication system is required to provide appropriate connection between the components and co-ordinate whole system toward enterprise goals [43, 44].

A decomposition approach breaks down the functionality of a system [19, 43]. A function identifies all activities that a system performs, the inputs they require and the outputs they generate. Also, a function (also called activity, operation, task, etc.) is recursively used at all level of modelling in

---

1 Arguably, the definition of the particular level of modelling may also address the instantiation of an individual product in a specific industry. In this case, the definition of the other level of generality will be affected and need to be redefined.

2 Considering the functionality of a system, one can call every active element a ‘function’. This includes terms such as: action, operation, task, activity and process. Vemadat [19] states that these terms refer to the different level of granularity of function. An action can be seen as something more atomic than a task or any activity, which is itself more atomic than a process.
any depth. This approach provides a hierarchical decomposition, which identifies a clear definition for functional responsibilities and attributes at a various levels of system abstraction. However, because this approach is time independent, it ignores the dynamic attributes of the system and cannot sufficiently handle a dynamic operation with variable resource allocations.

Another approach to modelling decomposition, so called ‘process modelling’, focuses on the ‘processes’ realised by the system [36, 53]. A complete life cycle of a process can be modelled by considering the flow of control, information and physical resource requirements regardless of the organisational aspects. A process is based on three essential items: task or activity; resource; and time. The most detailed part of a process is commonly called ‘activity’. A set of activities operated in a logical order compose the overall process model. There also may be a timed sequence attached to a set of processes [120, 182].

**Higher-order models** - As mentioned earlier, a modelling architecture may consist of a number of models, applications, or tools to provide appropriate control level over the system. Integration between the models within an enterprise can also be formalised in a logical way to promote internal communication and co-operation. The formalisation logic, which identifies correlation between the ordinary portions of manufacturing model (e.g. information model and function model) is called a ‘meta-model’ [177]. A meta-model prescribes structures in a restrictive manner (might be mathematical or rule-based) in which the models relate models to each other [183]. The example of such models within a manufacturing enterprise can be a model to define relationships between finance model and manufacturing model.

Furthermore, in a global modelling environment, some objects, entities, or ideas may be used in different models. Having a unified and shareable understanding from these entities in a specific domain will assist in the transition of that entity from one model to another. The ontology defines these common objects. The example of a common object defined in an ontology can be the definition of an ‘activity’ in a manufacturing enterprise. The combination of meta-models and ontology in a global modelling system is called ‘Common Model’ [183]. A common model provides a clear and comprehensive understanding of the system goals and perspectives.

Another high-order modelling concept is the notion of a ‘Reference Model’. There is no standard definition for a reference model, but the common point in different ideas about reference models is that:

"a reference model is a combination of frequently used tools, techniques, methods, concepts, reusable modules, common objects and so forth (or the knowledge about them) implemented in a specific domain of manufacturing environment" [39, 83, 176].

Using reference model simplifies the starting point of the modelling by offering knowledge about surrounding systems, restrictions and rules. Various reference models could exist in an overall manufacturing system to support different aspects of modelling. The degree of usefulness of a reference model varies depending on the quality and relevance of the knowledge it contains about...
the environment. Although these do not change the characteristic of a reference model, it could improve by repeated use and the accumulation of new knowledge.¹

**Principles of modelling** - Before beginning to model a system, a number of important issues need to be addressed. The reason for modelling and the benefit, which is expected to be gained, should be clearly defined [19, 183]. The likely extent of the modelling (i.e. how generic it is to be) and the detailing level or granularity needs also to be decided, as well as the perspectives which should be covered (i.e. function, information, etc.).

When designing a model, the modularity of its structure should be considered [28, 131]. Many systems can be broken down into sub-components or so called ‘modular building blocks’. These components may be reused by the model (i.e. reusability characteristic) and may also be designed in a generic manner. This improves the generality of the model. During the design of these components, integration issues between the components must be considered. Using well-defined (or standard) knowledge and data transaction methods and physical connection techniques (e.g. plug-ins) integration can be achieved in a flexible (i.e. readily modifiable) way. The flexibility of modelled components or the complete model is an important factor. The flexibility may be expressed in terms of resource allocations to the model, adaptability within the operating environment, organisational versatility, customisability for system developers and for the end-users (e.g. at execution level) [131] and so on.

The behaviour of components or system models is not the same as their functionality. A function is what the model carries out, disregarding behaviour, i.e. the way a function is carried out [28]. This provides a great deal of flexibility for the system designer to implement various method and tools to improve the model efficiency and reliability. This is also important to appreciate the separation of the data from the control or process [56].

The presentation of the model in terms of the logical language (syntax and semantic) and the way to visualise the models are significant parts of the modelling techniques. The model must clearly and extensively represent what it is supposed to model and visualise the system in an understandable way.

A model should be ‘complete’ [56] [86]. It should be able to handle various events, which may affect it in a real world implementation of the model. A deterministic design may jeopardise the viability of the model in a real world system.

¹ Some researchers describe the term reference model as "a configuration of components that each execute their own globally defined, distinct tasks but interact to realise the task of the system as a whole" [39]. In order to distinguish this definition from the term generic model, the author does not consider it to be an alternative meaning of the reference model in this work.
A number of modelling architecture, methods and tools were reviewed in chapter 2. Here some of these methods are considered in a greater detail as below.

For the sake of completeness, supporting literature collected and documented in this appendix is accompanied by some of the publicly available material related to this research, that were adopted and documented (as indicated) in the following sections.

G.4.1) CIMOSA Modelling Framework and Constructs

(The context of this section has been adopted from public domain material at [184])

The modelling framework shown in Figure G.2 structures the CIMOSA Reference Architecture into a generic and a partial modelling each level supporting different views on the particular enterprise model. The concept of views allows working with a subset of the model rather than with the complete model providing especially the business user with a reduced complexity for his particular area of interest. CIMOSA has defined four different modelling views Function, Information, Resource and Organisation. However this set of views may be extended if needed.

The CIMOSA Reference Architecture supports three modelling levels of the complete life cycle of enterprise operations (Requirements Definition, Design Specification and Implementation Description). Again, the sequence of modelling is optional. Modelling may start at any of the life cycle phases and may be iterative as well. Depending on the intention of model engineering, only some of the life cycle phases may be covered.
Enterprise operation should not be modelled as a large monolithic model but rather as a set of co-operating processes. With a set of common building blocks, the CIMOSA Reference Architecture provides the base for evolutionary enterprise modelling. This allows different people to model different areas of the enterprise but provides the integrity of the overall model.

Figure G.3 shows the basic set of common building blocks for business modelling. Processes, Events and Enterprise Activities are the object classes that describe functionality and behaviour of the enterprise operation. Inputs and outputs of Enterprise Activities define the information (Enterprise Object) and the resources needed. Organisational aspects are defined in terms of responsibilities and authorisation (Organisation Elements) for functionality, information, resources and organisation. They are structured in Organisational Units or Cells. CIMOSA employs the object-oriented concepts of inheritance, structuring its constructs into a hierarchy of object classes.

**Process Based Enterprise Modelling**

CIMOSA model engineering is demonstrated in Figure G.4 that shows three enterprise domains (DM 1-3) each one represented by its functionality - a set of Domain Processes. Domain Processes communicate with each other via Events and Results.

As shown in Figure G.5, decomposition of Domain Processes (DP2.1) via Business Processes leads to an identification of Enterprise Activities (EA1-5).

The connecting control flow represented by a set of Behavioural Rules is shown in Figure G.6. The network of these Enterprise Activities is the functional and dynamic representation of the Domain Process DP2.1. Events (1-2) relate to Domain Process DP 2.1 actually triggers Enterprise Activities EA1 and EA2 and are produced by EA3 and EA5 respectively.

The different Inputs and Outputs identified for each Enterprise Activity are shown in Figure G.7. All inputs and outputs are other constructs of the CIMOSA modelling approach. For those details reader should refer to CIMOSA references.

At the system design level Enterprise Activities are further decomposed into Functional Operations (Figure G.8, Part A). CIMOSA Functional Operations are defined in relations to their executing resource types; the Functional Entities. Each Functional Operation will be completely executed by one Functional Entity, but a Functional Entity may be capable to execute more than one type of Functional Operation. (Figure G.8, Part B) Functional Entities are resources which are capable to receive, send, and process and (optional) store information. They are therefore active resources of the enterprise such as humans, machines and applications (e.g. CAD systems, MRP systems, etc.)

**CIMOSA Integrating Infrastructure**

For model engineering and model driven enterprise operation control and monitoring, especially in heterogeneous environments the Integrating Infrastructure provides a set of generic IT service entities.
These service entities enable model engineering and model processing. Control on execution of the Implementation Description Model is provided by the Business Entity that receives Events and creates occurrences of the related Domain Process and all its contents. Business Process Control, Resource Management and Activity control (all part of the Business Entity) analyse the model content, assign the resources, identify the required information and connect to the necessary Information Technology Resources and the Manufacturing Resources via the Common, Information, and Presentation Entities. Ultimately, the Business Entity controls the execution of the Domain Process and the underlying network of Enterprise Activities, which represent the model.

CIMOSA Information Modelling

According to Jorysz/Vernadat [152] the information modelling paradigm used in the information view of the CIM-OSA architecture is a framework for the open modelling of enterprise
The information model at the Requirement Definition modelling level is based on the following concepts: enterprise objects, object views, information elements and an object abstraction mechanism.

An object view is a description of a particular aspect of an enterprise object. It is made of information elements and/or other object views. An information element is an indivisible piece of information at the atomic level of data of the CIM-OSA information view. Enterprise objects are information entities of the enterprise defined by attributes, which can be either information elements or a lower level of enterprise objects. The objects are linked together by means of object abstraction mechanisms which include: Generalisation which defines an object type as a more generic type of object; Aggregation which considers an object type as a conjunctive collection of sub-component objects (PARTOF link); Particularisation which refers to the linking an enterprise object to an enterprise object type (MEMBEROF link); and Generalised Relationships which refer to user-defined relationships between enterprise objects (LINKEDTO link).

At the Design Specification Modelling level, the entity relationship approach was selected by CIMOSA architecture to transform object-oriented information descriptions obtained at the previous modelling level, into a data-oriented model at this level. The underlying concepts of this approach have been constructed based on conceptual and external schemata of the three-schema approach using entity relationship attributes model. In this level of modelling a series of logical functions are built to transact on an information storage system, mainly relational database systems. The transactions on data storage system are constructed within conceptual schema and presentation of data and the user interface to access data are provided based on external schema.

When seeking to optimise conceptual and external schema, the system designer needs to utilise the CIMOSA information modelling methodology at the Implementation Description level. The SQL language is the basic language selected to communicate with database systems. The actual method of organising the conceptual schema used by the physical storage media is defined in the internal schema. The internal schema translates conceptual schema into a computer processable information model to provide an executable information model.

Application of CIMOSA

According to Zelm [137] CIMOSA has been studied, implemented and tested by a wide range of industrial and academic groups. Some of these groups are outlined below (mainly adopted from [137]).

- Fiat Auto/I - Gearbox Production [185],
- Magneti Marelli Ricambi/I - Automobile Components Supplier,
- Koehler/D - Paper Production [186],
- Traub/D - Machine Tools [187],
- ELVAL/GR - Aluminium Casting [188],
- Advanced Information Technology in Design and Manufacturing - ESPRIT Consortium AIT - Reference model for product development in the Auto/Aerospace Industry, developed by FIAT,
- Bottom-up Process Modelling and Simulation - ESPRIT Consortium PRIMA - Reference model for processes in the petro-chemical industry by University Karlsruhe - [170, 189],
- Decentralised Information and Communication Structures - The DARIF Project by Engineering School Offenburg [190],
- CIMOSA Organisational Aspects - by ITEM St Gallen University [137]
- Events and Exception Management - by EPFL Lausanne University [191].

For further details on the CIMOSA framework see [23, 56, 118, 181, 192, 193].

G.4.2) GERAM

(Context of this section has been adopted from public domain material at [62])

Starting from an evaluation of existing enterprise integration architectures the IFAC/IFIP Task Force on Architectures for Enterprise Integration has developed an overall definition of a generalised architecture. This proposed framework was labelled as GERAM (Generalised Enterprise Reference Architecture and Methodology).

GERAM is a generalised framework for enterprise integration and business process engineering. It identifies the set of components recommended for use in enterprise engineering. This set of components is identified in Figure G.9 and briefly described in the following. Starting from defined concepts to be used in enterprise integration (GERA), GERAM distinguishes between the methodologies for enterprise integration (GEEM) and the languages used to describe structure, contents and behaviour of the enterprise (GEML).

Enterprise modelling is seen as the major item in enterprise engineering and integration. Therefore, both the methodologies and the corresponding languages will be implemented in enterprise modelling tools (GEMT) which will support the enterprise integration process. Ontological theories (OT), generic enterprise models (GEMs) and generic enterprise modules (GMs) will support the modelling process by providing means for more efficient modelling. The modelling process will result in an enterprise model (EM) which represents all or part of the enterprise operation. These models will allow simulation of operational alternatives and thereby their evaluation leading to the optimum structure, contents and behaviour of the enterprise operation. **GERAM provides a generic description of all the elements recommended in enterprise engineering and integration.**
Definitions of GERAM Components

GERA—Generic Enterprise Reference Architecture: Defines the enterprise related generic concepts recommended for use in enterprise integration projects. These concepts include enterprise systems life cycle; business process modelling; modelling languages for different users of the architecture (business users, system designers, IT modelling specialists, others); integrated model representation in different model views.

GEEM—Generic Enterprise Engineering Methodologies:

Describe the generic processes of enterprise integration. These methodologies may be described in terms of process models with detailed instruction for each step of the integration process.

GEML—Generic Enterprise Modelling Languages: Define the generic constructs (building blocks) for enterprise modelling adapted to the different needs of people creating and using enterprise models.

GEMT—Generic Enterprise Modelling Tools: Define the generic implementation of enterprise-integration methodologies and modelling languages and other support for creation and use of enterprise models.
EM—Enterprise Models: Represents the enterprise operation. These models will be represented using generic modelling language constructs.

OT—Ontological Theories: Formalise the most generic aspects of enterprise-related concepts in terms of essential properties and axioms.

GEMs—Generic Enterprise Models: Identify reference models (partial models) which capture concepts common to many enterprises. GEMs will be used in enterprise modelling to increase modelling process efficiency.

GMs—Generic Modules: Identify generally applicable products to be employed in enterprise integration (e.g. tools, integrating infrastructures, and others.).

G.4.3) CIMBIOSYS Integrating Infrastructure

Work undertaken in the MSI Research Institute during mid to late 1980’s identified the need for an integration infrastructure that separates system functionality from system integration issues [138]. Major advantages of such an approach are: removing interaction knowledge like communication protocol, data structure and interaction mechanism from application by placing knowledge within integrating platform; dealing with complexity since applications need only to interact with integrating platform rather than other applications; assisting in creation of open application by enabling the utilisation of a consistent set of integration services [139].

The internal functionality of the CIMBIOSYS platform can be decomposed into four functional blocks as depicted in figure G.10. The functional blocks include:

**Integrating Infrastructure:** This is realised by inter-process communication and low-level handshaking functions required to interact with the integrating infrastructure.

**Application Event Management Functions:** These provide a consistent set of mechanisms for event management. Functional capabilities provided by these mechanisms can be accessed and used by the application code and are used by both application service and integrating infrastructure interfaces.

**Application Service Interface:** This is a suite of ‘C’ runtime functions, which facilitate access to integration services provided by the integrating infrastructure.

**Application Code:** This is code provided by application implementers that achieves the application functionality required in a given system.

For further reading on the CIMBIOSYS infrastructure see [68, 139].
G.4.4) SEWOSA

The SEWOSA CASE tool is a graphically based enterprise modelling tool created using a META-CASE tool. Each diagram of SEWOSA illustrates a level of modelling from requirement definition modelling level to the design specification level. There are associated textural description templates for each diagram. At the requirement definition modelling level, SEWOSA offers the following sets of diagrams to describe system capabilities, being organised into a specific domain; ‘Context diagram’ to define CIMOSA and non-CIMOSA domains and their relationships, ‘Domain diagram’ to specify domain processes of each domain, ‘Structure Diagram’ to decompose each domain process into atomic functional elements of business processes and enterprise activities, ‘Behaviour diagram’ to describe procedural rules related to the functionality of each domain process in terms of the flow of control, ‘Functional diagram’ to define the inputs and outputs of each element of domain processes in terms of the flow of material, information and control.

At the design specification modelling level, SEWOSA defines interactions among the components of the domain to determine how the system will operate. To achieve this SEWOSA uses the following diagrams: ‘Object diagram’ to specify the functional entities related to business entities in terms of functional operations and information entities and their relationships within an integrating infrastructure environment; ‘Activity Behaviour and Entity Behaviour diagrams’ to define operational procedures for each function entity and enterprise activity in terms of functional performance, this being based on use of a Petri-Nets technique; ‘Resource diagram’ to specify active and passive resource components and their relationships with functional entities which are able to execute a set of function operations; ‘Configuration diagram’ to specify the computer configurations of a system.

SEWOSA enables the definition of Stochastic time Petri-Nets models which can be used to analyse and simulate the dynamic behaviour of a system. Thereby it provides a rapid prototyping facility to test systems structure by executing and observing behaviour of the model. At the implementation description modelling level, emulated components of the system are replaced by physical elements, which are to be used in real world system. The SEWOSA workbench does not include a method for information modelling. For further reading refer to [117].
G.4.5) Modelling Framework Comparison

(Context of this section has been adopted from public domain material at [72])

A comparison of modelling methodologies is shown in Tables 1 through 3. GERAM has been used as a reference to classify the various parts of the different methodologies considered. The tables indicate the terminology problem existing in enterprise modelling. But there is a surprisingly high level of terminology consistence [72].

<table>
<thead>
<tr>
<th>GERAM</th>
<th>ARIS</th>
<th>CIMOSA</th>
<th>GRAI/GIM</th>
<th>IEM</th>
<th>PERA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>not defined</td>
<td>not defined</td>
<td>not defined</td>
<td>not defined</td>
<td>EBE Identification (Enterprise Business Entity)</td>
</tr>
<tr>
<td>Concept</td>
<td>not defined</td>
<td>not defined</td>
<td>not defined</td>
<td>not defined</td>
<td>EBE Concept Layer</td>
</tr>
<tr>
<td>Requirement</td>
<td>Operation Concept</td>
<td>Requirement Definition</td>
<td>Concept Level Analysis</td>
<td>Requirement Definition</td>
<td>EBE Definition Layer</td>
</tr>
<tr>
<td>Design -Detailed Design</td>
<td>EBE Identification</td>
<td>Implementation Description</td>
<td>Realisation Level - Technical Orient Des</td>
<td>Implementation Description</td>
<td>EBE Manifestation Layer</td>
</tr>
<tr>
<td>Implementation</td>
<td>(Operation)</td>
<td>Model Maintenance</td>
<td></td>
<td>EBE Operation Layer</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Decommission</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table G.1: Modelling Framework Comparison - Life cycle (Modelling Levels) – adopted form [72]

<table>
<thead>
<tr>
<th>GERAM</th>
<th>ARIS</th>
<th>CIMOSA</th>
<th>GRAI/GIM</th>
<th>IEM</th>
<th>PERA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Function View (static)</td>
<td>Function View (static)</td>
<td>Function View (static)</td>
<td>Information View</td>
<td>Info. Model View</td>
</tr>
<tr>
<td>Information</td>
<td>Data View</td>
<td>Information View</td>
<td>Information View</td>
<td>Info. Model View</td>
<td>not defined</td>
</tr>
<tr>
<td>Decision/ Organisation View</td>
<td>not defined Organisation View</td>
<td>not defined Organisation View</td>
<td>Decision View Physical View, Organisation View</td>
<td>not defined Human and Organis. Arch.</td>
<td></td>
</tr>
</tbody>
</table>

Table G.2: Modelling Framework Comparison - Model Views– adopted form [72]

<table>
<thead>
<tr>
<th>GERAM</th>
<th>ARIS</th>
<th>CIMOSA</th>
<th>GRAI/GIM</th>
<th>IEM</th>
<th>PERA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic</td>
<td>Generic</td>
<td>Generic</td>
<td>Generic</td>
<td>Reference</td>
<td>not defined</td>
</tr>
<tr>
<td>Partial</td>
<td>Reference Models</td>
<td>Partial</td>
<td>4 Levels of Abstract</td>
<td>Reference</td>
<td>not defined</td>
</tr>
<tr>
<td>Particular</td>
<td>Particular</td>
<td>Particular</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table G.3: Modelling Framework Comparison - Generality

252
Appendix

Collaborative Research Aimed at Extending the Modelling Life Cycle

Many research projects have considered the design, control and monitoring of manufacturing systems with different emphasis placed on particular life cycle phases. The semi-generic modelling of manufacturing cells reported in this thesis provides means of conceptually defining manufacturing cell control systems and specifies design methods for their construction, supporting these methods with implementation techniques that are used at the particular level of modelling. However, the approach developed in this study does not specify detailed system design in such a way that developed modelling descriptions define actual messages used to control and run physical devices in any cell.

Therefore with the purpose of gathering supportive evidence on the practicability of using semi-generic modelling concepts during the execution life phase, additional collaborative study was conducted by the author. Here the possibility of extending the life coverage was investigated. This appendix outlines how ongoing collaboration between the MSI Research Institute (at Loughborough University, UK) and the Mechanical Engineering Department (at Navarra University, Spain) is seeking to unify aspects of two PhD studies: one by the author, the other by Aguirre who developed a set of tools referred to as MCS Tools [194].

This research covered developed generic modelling concepts based on the combined use of CIMOSA and component-based systems engineering concepts. Part of this work developed modelling specifications for the particular level of modelling cell systems in terms of information and function entity relationships, and software component capabilities. Whereas Aguirre’s research developed modelling methods, tools and techniques to facilitate interaction and control amongst manufacturing devices by developing the so called MCSARCH architecture [194]. The prime aim of MCSTOOLS is to generate MCCS modules, called MMS servers that can be interconnected by means of a CORBA (“Common Object Request Broker Architecture”) integrating infrastructure. The structure of MCCS modules has been defined along with their use of the Manufacturing Message Specification (MMS) standard (ISO 9506). MMS servers are designed as CORBA object components capable of realising the functionality of elements defined by the COSIMA conceptual model [195]. This model has five main cell control functions, namely: scheduler, dispatcher, mover, producer and monitor.

A number of potential links could have been established between the two PhD outcomes. The links include: a) link high-level functional models (from MSI) to executable objects at the operation level (from Navarra) to extend the life-cycle coverage; b) specify and develop generic means of realising consistent monitoring systems capable of collecting dynamic data at the execution level.
and analysing and reporting on overall system performance at the control level; c) integrate generic information modelling and data manipulation methods, used at the execution level.

MSI's work provides means of specifying particular jobs, i.e. by a "job operation" activity (the key activity §7.6) and transforms the operation descriptions defined in a modelling environment into the real world resource execution codes. This provides "hooks" between a high-level modelling structure and an implementation approach. Correspondingly MCSTOOLS allocate physical resources to jobs, translate functional descriptions into sets of machine code, and executes the physical devices.

MCSTOOLS uses STEP part AP213 (ISO/DIS 10303-213) as the standard for data interchange, that is related to information contained within product process plans. Thereby MMS servers are able to execute process plans defined in conformance with the STEP AP213 ("Numerical Control Process Plan Definition Data Standard") [196]. This program defines activities and information flow. The accomplishment of an "activity" is the execution of a method defined in the public interface of the MMS server. Thus, these activities can be MMS messages or non-MMS messages previously defined within the structure of the MMS server. Furthermore, definitions provided by the STEP AP213 standard allow to define plans of activities that need not necessarily to have a direct relationship with NC process plans, but could also have a wider use. In this way, STEP AP213 can extend its semantic and thus, it is possible to define a great deal of plans of activities in a manufacturing system. Using such a capability, MMS server behaviour can be defined in two ways, including: a) internal programming and b) execution of STEP AP213 files, which define specific behaviour for the MMS server. With b) it is possible to define dynamic behaviour for the MMS server, i.e. external modules can define specific behaviour of the MMS server by sending files coded in the STEP AP213 standard. Option is (b) offers possible link to higher modules that have a capability to define the behaviour of an MMS server.

Further research is required to investigate solutions to problems of integrating MSI's conceptual/design approach and Navarra's implementation/execution approach. Currently the compatibility of the two modelling architectures needs further investigation but underlying concepts of the two approaches have been mapped.
Appendix

I

Publications


   Abstract: The need for agile manufacturing systems requires new enterprise modelling methods and tools which support their (re)configuration and (re)engineering. This paper describes how an enterprise modelling toolset based on the CIMOSA enterprise modelling framework and constructs has been extended to structure and support information modelling based on EXPRESS and STEP. Furthermore the paper describes use of this extended toolset to define and use a semi-generic model of manufacturing cell control systems, which is applicable primarily in printed circuit board manufacturing domains.


   Abstract: A new approach to the design and reconfiguration of change capable manufacturing cells is described. The approach is based on (i) the development of particular models of cells, where the use of CIMOSA modelling constructs is structured and informed by a semi-generic model of similar manufacturing cells and (ii) the use of new constructs and tools that operationalize particular models in the form of an explicit, model-based configuration of cell resources and software components. The paper describes key elements of the semi-generic model and a case study application of the approach when designing and prototyping a case study manufacturing cell.


   Abstract: Computer Integrated Manufacturing (CIM) systems with a significant level of human-computer interaction are often inefficient. This is particularly problematical for those users who have to interact with multiple subsystem interfaces. These difficulties can be traced back to the fact that representation of the user in existing manufacturing models and systems is inadequate. An approach that increases user representation to improve CIM interface design is proposed, in which stereotype-based user and task models are used to specify a common user interface for each individual system user. An overview of the architecture is followed by discussion of an application domain (statistical process control) in which a demonstrator based on the architecture has been tested.


   Abstract: Computer Integrated Manufacturing (CIM) systems are complex in terms of performing a variety of activities, maintaining a range of information, and involving various classes of users with differing levels of knowledge and skills, and different levels and time spans of decision making. Much investment and effort has been made to formalise and automate the performance of the CIM elements in a manufacturing system. However, each CIM sub-system will typically have its own terminology, procedures and presentation formats. This places a heavy and unnecessary burden on users, resulting in frustration and reduced effectiveness.

   Research has been carried out by the Manufacturing Systems Integration (MSI) Research Institute at Loughborough University towards the generation of a user-oriented interface for CIM systems. This research has resulted in a conceptual approach, which incorporates a generic user/task model, which enables the generation of flexible and reusable software components to form a semi-generic user interface for CIM users. The CIM user interface provides presentation tools to monitor and control the performance of the CIM elements.

   Advanced modelling and integration technologies have been deployed to enable the system to cover a wide area of manufacturing domains. These technologies include modern manufacturing modelling architectures such as CIMOSA [1] and GERAM [2], and advanced communication techniques such as those used by Web-based software applications in manufacturing environments. The implementation issues of the generic user interface concept, along with its application within an industrial case study are discussed in this paper.
Abstract: Contemporary approaches to the design and construction of manufacturing systems often result in inflexible enterprises which cannot readily be sharply tuned to changing business goals. Manufacturing cells represent a typical domain in which the existence of hard, inflexible links between tasks and resources will result in sub-optimal performance and an inability to cope with change. Hence this paper reports progress on the definition of a semi-generic meta model of manufacturing cell control systems which can be used in conjunction with (a) modelling tools and (b) computerised infrastructure technology to facilitate system reengineering and reconfiguration and the reuse of cell components in different manufacturing cells. The meta model has been conceived, formalised and evaluated under laboratory conditions by utilising systems modelling and integration tools, based on CIM-OSA, Petri nets, EXPRESS, STEP and CIM-BIOSYS infrastructure elements.

Abstract: Ongoing research being carried out in the Manufacturing System Integration (MSI) Research Institute at Loughborough University is aimed at studying the use of human factors in generating user interfaces for CIM systems. The framework of this research includes an investigation of CIM system users, what tasks they perform, and what those user require from a computer interface. Within this framework, a methodology is being established to support communication between users and interfaces by using a CIM user model to help construct a generic and modular task-oriented user interface. Communication between the user interface and other manufacturing models is specified via a well-defined interaction model.
This paper discuss the application of agent-based technology to support this interaction model. AN overview of the general context of research is followed by discussion of the implementation of agent-based techniques, along with details of a standard knowledge exchange format.

Abstract: The background research undertaken at the Manufacturing System Integration(MSI) Research Institute at Loughborough University into the development of a common user interface for Computer Integrated Manufacturing (CIM) systems based upon a generic user model for CIM is discussed in this paper. Issues such as human computer interaction, user modelling and interaction model are outlined along with a discussion of user interface enabling techniques. Implementation issues surrounding the employment of intelligent software agents and Knowledge Query Manipulation Language (KQML) are detailed.
A case study based upon CIM interaction model involving a manufacturer of pharmaceutical punches and dies is briefly discussed.


78. W.A. Scheer, C. Kruse, "ARIS-Framework and Toolset - A Comprehensive Business Process

259


[84] QCIM, "Integrated Enterprise Modelling - Construct Specification", Contribution of DCIM (Germany) to CEN TC 310/WG 1 IPK Berlin (Version 1.0), Berlin, Germany 1993.


Part 2. Information


[190] Darif,, 1997


Index

A
Activity Behaviour Diagram ........................................ See SEWOSA
Agility ................................................................. 17, 19, 20, 23, 32, 167
ALBERT ................................................................. 48
Allocated Procedural Rules ........................................ See Procedural Rules
ARIS ........................................................................ 42, 49, 252

B
BE ........................................................................ See Business Entity
Behaviour Diagram ................................................ See SEWOSA
BP ........................................................................ See Business Process
Building Block ......................................................... 155, 172, 173, 176, 179
Business Entity ......................................................... 80, 97, 109, 125, 150, 172
Business Process ....................................................... 88

C
CATools ................................................................... See Cell Application Tools
CDTool ...................................................................... See Cell Design Tool
Cell ............................................................................ 35
Cell Controllers Definition ........................................ 35
Cell Grouping ........................................................... 82, 110, 112, 126, 144
Current Problems Reported ........................................ 57
Manufacturing Cell Definition ...................................... 33, 237
Typical Tasks ............................................................ 33
Cell Application Tools ................................................ 131
Cell Design Tool ....................................................... 130
Change ..................................................................... 158
Business Change ....................................................... 158
Change Capability ..................................................... 17, 136, 157, 171, 178
Change in Business Requirement ................................. 23, 24, 136
Change in Operational Details .................................... 53, 136, 151, 161, 163...
Change in Production Requirements ............................ 23, 24, 64, 136, 160
Information Requirements Changes .............................. 161
Integration and Networking Systems Changes ............... 160
Process Changes ........................................................ 160
System Engineering Changes ...................................... 159
CIMBIOSYS
Applications ............................................................. 200
Codes ...................................................................... 203
Definition .................................................................. 49
Integration System ..................................................... 133, 250
CIMOSA
Architecture ............................................................. 44, 54, 71, 85, 106, 168, 207, 247
Business Model ........................................................ 245
CIMOSA Information Model ........................................ 93, 246
CIMOSA-EXPRESS Mapping ...................................... 103
Consortium ................................................................ 37, 76, 92, 122
Constraints .................................................................. 54
Definition .................................................................... 44, 244
Industrial Applications ............................................... 247
Information Model ..................................................... 93, 246
Integrating Infrastructure ........................................... 245
Modelling Framework ............................................... 244
Modelling Views ....................................................... 85
Non-CIMOSA Domain ............................................... 86, 87, 88, 105
Rationale to Choose in this Study .................................. 54
CIM-Tool .................................................................. 50
Component
Component-Based Approach ....................................... 51, 72, 105
Software Component ............................................... 109, 111, 177
Software Component Templates ................................. 117, 118
System Component ................................................... 116, 117
Types of Components in this Research ......................... 107
Conceptual Design Level ........................................... See Modelling Level
Configurability ......................................................... 36, 68, 78, 96, 133, 150, 160, 164, 165

167, 171
Connectivity Object .................................................... See SEWOSA
Context Diagram ...................................................... See SEWOSA
CORBA .................................................................... 38, 48, 110, 133, 161, 253
Core Functions .......................................................... 78

D
Data Model ............................................................. 41, 43, 55, 93, 201
Database Management System ....................................... 145, 163, 190, 202
DBMS ...................................................................... See Database Management System
Detailed Design Level .............................................. See Modelling Level
Domain Process ........................................................ 87
DP ........................................................................... See Domain Process

E
EA .......................................................................... See Enterprise Activity
Enterprise Activity ...................................................... 88
Enterprise Integration .................................................. 37, 72, 244, 248
Enterprise Modelling26, 39, 43, 44, 51, 85, 238, 245, 249
Enterprise Modelling - Definition26, 39, 43, 44, 51, 85, 238, 245, 249
EO ........................................................................... See Enterprise Object
Estelle ..................................................................... 106, 169
Expandability ............................................................ 165, 171
EXPRESS
Attribute ..................................................................... 47, 103
CIMOSA-EXPRESS Mapping ...................................... 103
Code ....................................................................... 188
Entity ...................................................................... 47, 103
EXPRESS-G ........................................................... 105, 182
Language ................................................................... 47
Schemata .................................................................. 47, 103, 104

F
FE .......................................................................... See Functional Entity
FO .......................................................................... See Functional Operation
Function Diagram ..................................................... See SEWOSA
Functional Entity ..................................................... 97, 99, 112, 126, 245, 246
Functional Operation ................................................... 97

G
Generality ............................................................... 68, 115, 164, 241, 252
Generic Model ........................................................... 76, 77, Also see Model
Generic Task ............................................................. 82
GERAM ................................................................. 43, 55, 71, 169, 248, 249, 252
GRAL/GIM ............................................................. 42, 43, 252

H
Holonic Manufacturing ............................................... 51, 83, 159

I
IDEF ........................................................................... 46
IE ........................................................................... 43, 252
Information Entities ..................................................... 85, 173
Information Model ...................................................... 91
Infrastructure Services ............................................... 63, 127, 133, 153
INGRES .................................................................. 129, 177, 190, 191, 202, 203
IPSYS ......................................................................... 190, 191, 207, 212
ISO ........................................................................... 47, 49, 51, 54, 253, 254
IThink ........................................................................ 50

J
Job Set .................................................................... 137

K
KARE Project .......................................................... 169
Key Activity ............................................................. 138, 254
KIF ........................................................................ See Knowledge Interchange Format