Sheffield Hallam University

Author:

Title:

Degree:

Year:

Copyright Declaration

Date	Name and address (in block letters)	Signature
17/8/2007	LEE KOK CHONG	[signature]

METHODOLOGY FOR INPUT DATA MODELLING IN THE SIMULATION OF MANUFACTURING SYSTEMS

Kapila Nisanji Hakmana Parana Liyanage

A thesis submitted in partial fulfilment of the requirements of Sheffield Hallam University for the degree of Doctor of Philosophy

SHEFFIELD HALLAM UNIVERSITY
March 1999

PREFACE

This thesis is submitted to the School of Engineering of Sheffield Hallam University for the degree of Doctor of Philosophy. This study was conducted in the division of Design and Manufacturing of School of Engineering.

I would like to express my sincere gratitude to my Director of Studies Dr T. D. S. Perera for his guidance, encouragement and support throughout the course of this research. I would also like to thank my second supervisor Dr W. M. Hales for his help and support.

I also thank all the members of the Modelling and Simulation Research Group; especially Kevin Campbell, Ben Tye, David Clegg, Matthew Loynes and Frank Schormann for their help and support. Also my colleagues and administrative staff within the school of Engineering, and the Research office for their help and support. Finally, I like to thank the School of Engineering of Sheffield Hallam University for awarding me a scholarship to undertake this research programme.

The results obtained during the course of this research are to the best of my knowledge original, except where reference is made to the work of others.

K.N.H.P. Liyanage

ABSTRACT

Computer simulation is a well-established decision support tool in manufacturing industry. However, factors such as wrong conceptualisation, inefficient input data modelling, inadequate verification and validation, poorly planned experimentation and lengthy model documentation inhibit the rapid development and deployment of simulation models. A serious limitation among the above factors is inefficient data modelling. Typically, more than one third of project time is spent on identification, collection, validation and analysis of input data.

This study investigated potential problems which influence inefficient data modelling. On the basis of a detailed analysis of data modelling problems, the study recommends a methodology to address many of these difficulties. The proposed methodology, discussed in this thesis, is called MMOD (**M**ethodology for **M**odelling **O**f input **D**ata). An activity module library and a reference data model, both developed using the IDEF family of constructs, are the core elements of the methodology.

The methodology provides guidance on the best way of implementation and provide a tool kit to accelerate the data modelling exercise. It assists the modeller to generate a customised data model (entity model), according to the knowledge gained from the conceptualisation phase of the simulation project. The resulting customised data model can then be converted into a relational database which shows how the entities and relationships will be transformed into an actual database implementation. The application of the MMOD through simulation life cycle also enables the modeller to deal with important phases in the simulation project, such as system investigation, problems and objective definitions and the level of detail definitions. A sample production cell with different level of detail has been used to illustrate the use of the methodology.

In addition, a number of useful methods of data collection and the benefits of using a MMOD approach to support these methods and data rationalisation which accelerates the data collection exercise are also covered. The aim of data rationalisation is to reduce the volume of input data needed by simulation models. This work develops two useful data rationalisation methods which accelerate the data collection exercise and reduce the model complexity.

This work produced a novel approach to support input data modelling in simulation of manufacturing system. This method is particularly useful when the complex systems are modelled.

CONTENT

CHAPTER 1

INTRODUCTION

1.1 Simulation

Simulation is one of the most powerful analytical tools that can significantly facilitate the problem solving and decision making process. The definition of simulation is given by Pegden et al (1990) as "the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system". Computer simulation, because of its inherent capability to observe the system without the necessity of experiments with actual system has found widespread use as a decision support tool in the manufacturing industry, business systems, computer systems, chemical industry, communication networks, construction, transportation services and systems, health care systems services and military sectors.

1.1.1 Discrete-event simulation

Discrete event simulation is concerned with the modelling of a system by a representation in which the state variables change at sudden distinct events. A customer arriving at a bank and parts arriving and leaving at a workstation are examples of a discrete event. Most manufacturing systems are discrete-event systems. In this study we concentrate on only the discrete event simulation. There are several other types of simulation (Law & Kelton, 1991), and the main types of simulation can be distinguished on the basis of changing the state of the system over the time. In this thesis, the term "simulation" is used to refer to discrete-event simulation.

1.1.2 Simulation in Manufacturing

Manufacturing system simulation is one of the largest application areas in the simulation. Traditionally, simulation was mainly used in capital intensive projects such as the design of new factory layouts. Computer simulation is now seen as an integral tool in the design, planning, operation and restructuring of manufacturing systems (Wu 1996, Law and McComas 1997, Williams and Narayanaswamy 1997). Simulation software has improved greatly over the past few years and is becoming more common in manufacturing environments. Nowadays, most modern simulation software provides greater modelling flexibility, interactive animation and advanced integration facilities, making it easier for engineers and managers to use as a valuable tool for analysing manufacturing systems.

1.2 Problems Associated with Simulation

Simulation is an ever growing area of interest for many industries. It is obvious that the availability of affordable and user-friendly software tools have improved the usability of computer simulation and it is frequently used to address a wide variety of operational problems. Thus, the ability to construct simulation models quickly and effectively is far more important than ever before. However, the construction of a simulation model is just one part of the overall simulation project effort. It can normally be split into the following steps :-

- Conceptualisation
- Data collection and analysis
- Model construction
- Verification and validation
- Experimentation and analysis of results
- Implementation and documentation

The main problems facing the effective use of the simulation are poor conceptualisation, inefficient input data modelling, insufficient model verification and validation, poorly planned experimentation and poor implementation and documentation. These factors can be considered as direct causes for excessive modelling time, unrepresentative models and inaccurate results.

1.2.1 Poor Conceptualisation

The conceptual model produces a complete specification of the model to build. The conceptual model phase is also known as model formulation. During the conceptualisation phase, the vital aspect is to determine the objective(s) of the simulation model. At this stage, the modeller also decides what system elements should be included in the model and what level of detail should be represented. The most difficult aspect of conceptual modelling is often considered to be the problem of choosing the appropriate level of detail. Poor conceptualisation misleads the computer model since the conceptual model is translated into a computer model.

1.2.2 Inefficient input data modelling

Input data modelling is a major activity in simulation projects. This activity encompasses the identification of required data, gathering of data, analysing and organising data and validation. Typically, these tasks are tedious and time consuming, specifically, when a large amount of data is involved in a model, which would be extremely difficult to handle. Unfortunately, there are no systematic methods for identifying, collecting and maintaining data. It has been observed that even in similar simulation projects, different practitioners adopt different approaches to identify, gather, analyse and organise necessary data.

1.2.3 Insufficient model verification and validation

Law and Kelton (1991) argue that one of the most difficult problems in simulation is to determine whether a simulation model is an accurate representation of the actual system being studied. Verification is the process of testing whether or not the computer program of the simulation model and its implementation is correct. Validation involves the process of comparing whether the conceptual model is an accurate representation of the system under study. However, model verification and validation are critical in the development of a simulation model. Sargent (1996) argues that unfortunately, there are no systematic tests that can easily be applied to determine the "correctness" of the model.

1.2.4 Poorly planned experimentation

Experimentation in simulation can be defined as the actual running of experiments and analysis of the results. The proper experimentation analysis is one of the most important aspects of any simulation study. It deals with issues such as (Robinson and Bhatia 1995, Shannon, 1998)

- determination of running times (length of the simulation run),
- determination of number of simulation runs (number of replications) to achieve a given confidence level,
- determination of the starting conditions (warm-up period),
- selecting the actual experiment that needed to be performed with alternative parameters, and
- determination of statistical test.

Prior to experimentation design, sufficient care should be given on these issues to ensure the accuracy of output data. Poorly planned experimentation may lead to the inaccuracy of the output result.

1.2.5 Poor implementation and documentation

The final two elements of any simulation project are implementation and documentation. In order to effectively depict the simulation results it is important to implement the output result in an effective manner. In particular, the capability to compare data from different simulation runs. During the documentation, all components of the model such as parameters, inputs, and outputs are well documented as well as the model itself. It makes the function of the model clearer to all who are considering it for reuse. It also provides insight as to where modifications might be needed to better fit the model into current application.

A serious limitation among the above factors is inefficient data modelling. Most of the simulation practitioners argue that in a typical model building exercise, data identification, gathering, analysis and validation can take more than one third of the project time. In fact, it seems that the effort required to model data has not significantly changed over the last decade.

In the 1980's limitations of simulation software led to complex features being frequently ignored from the modelling process; consequently, projects required less data. However, data had to be collected by manual means. Advances in simulation software have enabled the modellers to build more complex models in the 1990's, requiring large volumes and variety of data. As a result, the effort required to collate and analyse data remain somewhat the same.

1.3 Types of input data

The data required for a manufacturing system modelling application can be broadly varied according to the system under investigation and the types of project objectives. The manufacturing systems consist of a variety of part types and process routes, resources storage and material handling devices. These objects often require a variety of data types for simulation modelling as briefly described in table 1.1.

Objects	Data Items
Parts	part identification details, production quantities, assembly planning details, Due dates, arrival rates to the system, routing, etc.
Resources (Machine or People)	resources identification details, capacity resource reliability data such as pattern of breakdowns of machines, time to repair machine, etc.
Material Handling Devices	material handling devices identification details, capacity, speed, acceleration, declaration,
Process	process identification, process description and times, set-up details, rejection rates and rework details, machine loading/unloading details, etc.
Transportation	path identification details, distances, travel times, etc.
Storage facilities	storage identification details, capacity, etc.
Fixtures/Pallets	identification details, size, etc.
Scheduling	resources and material handling scheduling data (available & unavailable times), shift patterns, etc.

Table 1.1. Types of input data

1.4 Process of Modelling Input Data

In a simulation project, the process of input data modelling commences at the beginning of the study and it mainly encompasses the identification, collection, analysing and organisation and validation of data (see figure 1.1).

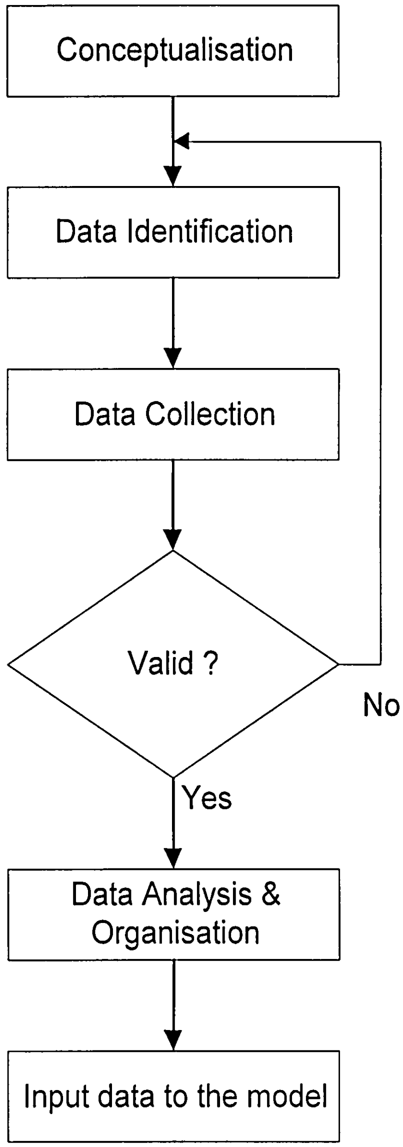


Figure 1.1 Steps in input data modelling

Step 1. Identification of input data

First, it is necessary to identify what types of data should be collected. This task is dependent on what types of system elements have been included in the model and what level of detail has been represented.

Step 2. Collection of input data

Once the data has been identified, it is necessary to collect it by reading available data sources, interviewing experts, directly observing the systems and making intelligent guesses.

Step 3. Validation of input data

During the data modelling, sufficient care must be taken to determine the accuracy of the data that is being used.

Step 4. Organisation and analysis of input data

The collated data must be recorded and it can be seen that often the collated data is stored in propriety formats. When the core data is altered, it may be necessary to analyse and organise the data for simulation models according to the user requirements.

Step 5 Input data to simulation model

The final stage of data modelling is inputting data to the simulation model. The required data for the simulator can be entered in manually through data-entry windows, from spreadsheets or databases.

1.5 Focus of the research

Much research has focused on various areas of simulation modelling but no attempt has been paid to the development of systematic approaches for input data modelling. Any reduction in the time taken for input data modelling will enable the practitioners to build simulation models quickly. The aim of this research is to reduce the amount of time needed for the stages of input data modelling in the simulation of batch manufacturing systems.

To accomplish this objective, it is necessary to :-

- understand and recognise potential problems associated with input data modelling exercise. How they impact on the inefficient data modelling and how to address them.

- investigate and develop a coherent methodology to support the steps in input data modelling and guidelines to demonstrate how the methodology can be applied to address the problems.

1.6 Outline of the Thesis

The literature covering the research background is reviewed in chapter 2. An analysis of data modelling problems, together with proposed research methodology, is discussed in chapter 3. It established background for research methodology is presented in this thesis. The proposed Methodology for Modelling of Data (MMOD) is presented in chapter 4, and it contains an activity model library (phase 1), a comprehensive reference data model (phase 2) and mapping tables (phase 3) to integrate both activity and data models. The method of generating customised data models for a given simulation project, i.e. the application of MMOD through a simulation life cycle is described in chapter 5. The data collection methodology for MMOD and some useful data rationalisation techniques to accelerate data collection is discussed in chapter 6. Finally, chapter 7 presents contributions of this thesis and further research areas.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Chapter 1 outlined the stages involved in the modelling of input data and concluded that further work is required to develop a methodology for systematic collection, analysis and representation of input data.

Prior to the programme of research is established it is necessary to review the work carried out in simulation and related area.

The key objectives of the review are to :-

- Assess different authors' views regarding inefficient data modelling
- Identify typical pitfalls which influence longer data modelling time
- Review current research works on input data modelling in the simulation

2.2 Different viewpoints of authors regarding the input data modelling

It is obvious that the development of simulation models is delayed when the correct data is not available in the right format at the right time. In the most real industrial applications reported in the literature, the model builders have raised a variety of issues surrounding the input data collection. On the basis of a number of industrial applications, Trybula (1994 & 1995) suggests that in a typical model building exercise, each phase may consume the following proportions of the project time.

Problem definition	~10%
Problem analysis	~10%
Model development	10% to 40%
Data gathering and validation	**10% to 40%**
Model verification and validation	~10%
Model experiments	10% to 20%
Analysis of results	~10%
Conclusion and recommendations	~5%

As outlined in Table 2.1, a number of other authors also argue that the time spent on data modelling can be excessively long.

Year	Author	Views
1981	Markowitz	Data collection and analysis take a long time. In many studies it takes longer time to gather and analysis data rather than designing and programming.
1990	Hatami	Collection and compilation of data for simulation may appear to be tedious or time consuming.
1992	Dietz	Simulation model can be only as good as its input data : " garbage in, garbage out". However, collecting data is time consuming and expensive. This expenses can be minimised by up-front planning and by collecting data intelligently.
1994	Trybula	Development of simulation models is delayed when required data are unavailable. Lack of data has turned many simple, short simulation projects in to ones of extraordinary duration.
1995	Lung	The duration for developing the simulation model can much depend on the amount of quality of data that has been collated from the client. If client data is not collated effectively it can be a major hindrance in trying to establish an accurate model.
1995	Robins& Bhatia	Data collection and analysis may take some time and therefore it is often performed in parallel with the other modelling activities.
1997	Les Oakshott	The collection of data for a simulation model is probably the most important part of a simulation project. If insufficient care is taken over this process the whole project can be a failure.
1998	Matt Rohrer & Jerry Banks	Simulation analyst spends less than 50 percent of the time actually building the model. The other time is spent collecting and structuring input data, writing specification and reports, experimenting with the model, and presenting results. Collecting and managing data can be a tremendous process.

Table 2.1 View points on data collection in simulation

2.3 Problems in input data modelling

Despite recent advances in simulation software, Trybula (1994) argues that the time to required to create a simulation model has remained virtually unchanged. According to his experience in the 1980's, when the first of the newer type of simulation language appeared, the time needed to create a model ranged from 6 weeks to 6 months or even a year. One possible reason may be the development of more complex models and the explosion of data required. In the early days, simulation systems could model only a limited range of features, hence, data requirements were minimal. Even then, data modelling was time consuming due to poor data organisation and paper based systems. As simulation software vendors introduced new modelling capabilities, the model builders began to model more complex simulation models. Although data is now more organised and computerised data management systems are available, the effort required to model data may remain more or less the same due to an increase in the required data types, volumes and other external factors. Consequently, it is necessary to look at other external factors which can influence the inefficient input data modelling through the different view points of the entire project.

A range of publications have been surveyed in order to investigate the issues related to the input data modelling problems and categorise these under the key stages of a simulation life cycle.

2.3.1 Initial Phase of the Simulation Project

Robinsion & Bhatia (1995) suggested that the initial phase of a simulation project is problem definition. This involves the understanding of a problem to be solved and the gathering of all necessary information. Once the problems have been formulated, the next stage is to set the objectives. Law (1990) explained that the level of model detail should depend on project objectives. Novels (1992) pointed out that many projects start without proper objectives. This aspect will have an enormous impact on the scope of the model, the level of detail to be incorporated and therefore the data that needs to be collected. Hatami (1990) explained that the level of detail required for a model can determine data requirements for the model. Furthermore, Robinsion & Bhatia (1995) have stated that data requirements can quickly be identified from the elements in the conceptual model and the level of detail explain exactly what data is required for the elements. According to the above discussion, it is evident that even at the initial stage of the simulation project, problem formulation, objective definition and determination of the level of detail influence each other and can have a significant impact on the input data modelling.

2.3.2 System Investigation

To build a valid simulation model, the modeller must have a good understanding of the system to be modelled. At this stage, it is important to determine the elements of the system which should be included in the model. However, identifying the system elements is not an easy task when the system is complex. Evans et al (1994) defined factors for the complexities associated with the modelling of advanced manufacturing systems (AMS). He argues that advanced manufacturing systems are complex because:-

- a wide variety of parts are produced by the system, and
- a number of different resources interact with each other in a complex manner.

According to this definition, a wide variety of parts increases the many different routes and this may lead to model decisions regarding the sequencing and scheduling of parts. He also mentioned that the types of resources associated with an AMS include pallets, fixtures, tools, robots, machines, conveyors, AGVs, operators, automated storage/retrieval systems and inspection, etc. John Carsion, Jerry Fox/Stephen Halladinn, Kenneth Musselman and Onur Ulgen [at panel discussion with Law A.M (Law, 1993)] pointed out some issues like a complex model needs more data, it is difficult to decide the level of detail for large complex models. According to the above discussion, it is clear that data modelling exercises may be difficult when the system is too complex.

2.3.3 Data Collection

Data collection is a complex issue throughout the entire project life cycle with many obstacles as described in the previous sections. In some cases, required data types having been identified , a significant amount of time is invested in collecting data.

There may be in some situations, if data is not available (Oakshott 1997, Robinsion 1994). This may typically happen when the simulation is for new systems. However, even an existing system can face the same problem due to poor data availability, missing data and impossibility of data collection.

Dietz (1992) has identified the most important steps of the simulation project and amongst his explanation, one key step is the "identification of input data sources". John Carson explained at a panel discussion which was conducted by A. M. Law (Law, 1993), that Obtaining accurate data is more problematic. They listed commonly frequently used cited data sources: computer databases, automatic data collection devices, maintenance records, production records, interviews and equipment specifications. Pegden et al (1990) state that four potential sources of information.

- Documentation
- Interviews
- Observation and Measurement
- Participation

However, he suggests that these potential sources may not provide 100% accurate information. The findings above provide evidence that inaccuracies can occur when data is collected from the sources.

2.3.4 Model Construction

According to Hlupic's (1993) simulation software evaluation, it is obvious that input data handling capabilities such as data storage, organisation, retrieval and manipulation facilities of currently available simulation software is not very high. However, Bank and Gibson (1997) suggest that many software vendors have included a facility to integrate with third party software such as modern databases and spreadsheets packages to provide this capabilities.

It can be seen from the above study that no attempts have been made to identify and properly document the spectrum of data modelling problems. It can be seen that only some underlying factors have direct impact on input data modelling exercise. However, these factors can influence the inefficient data modelling in hidden form. Therefore, in chapter 3, we will document the input data modelling problems in detail which are synthesised from above study. An analysis of results obtained from the questionnaire survey conducted at the 1997 Winter Simulation Conference (WSC) will also be presented to validate the identified data modelling problems.

2.4 Data Modelling Research in Simulation

There are very few researches that have been done in the area of data modelling in simulation. It appears that a lot of research based on the sub heading, "Input modelling" has focused on statistical data analysis techniques (Ex. Law et al, 1994, Wilson, 1997, Cheng 1993, Vincent & Law 1993 and Leemis 1996). The main objectives of these papers were to give guidelines on how to represent an appropriate probability distribution, select a distribution when data is unavailable and the analysis of inter arrivals, etc.

Joint Data Base Element (JDBE) research is one of the major researches in the area of input data modelling. It was developed to support modelling and simulation projects, especially for combat development at U. S. Army Electronic Proving Ground (Cole &

Valentine, 1993, McDonald and McDonnell, 1995). JDBE is a project for the sharing of modelling and simulation data by mapping existing databases into a standardised information model. Using the IDEF1X data modelling language as a basis, JDBE used reverse engineering principles that focus on information that is already supporting modelling and simulation applications. By standardising the information that is being used, JDBE directly enhances the potential for exchange of systems or restructuring of databases.

The JDBE project was carried out to capture the logical data structure for the existing multiple databases (e.g. training or analytical simulators, weapon systems, system testing, and decision support systems) which are independent of each other and non-standard. This may provide data that is eventually needed in various user information systems that already supported the modelling and simulation. They used reverse engineering methodology to describe existing databases. Because of this reverse engineering focus in the data, the JDBE project requires very little process modelling. Then, by grouping data according to the subject area and applying a data integration methodology, these logical data models are derived in the graphical IDEF1X data modelling language for existing multiple databases and merge into an integrated data model. The data also provides the meta data (data about data) needed in the US Army. The JDBE integrated data model has been created with the aid of ERwin CASE software tool and this software automatically generates the Structured Query Language (SQL) for database implementations. The JDBE team suggest that their methodology can assist organisations to integrate diverse data sources into the shared use of common databases to supply data for multiple applications. However, the JDBE approach does not cover all aspects of input data modelling in simulation.

Lung (1995) has developed a consistent approach to improve simulation project lead time. He argues that activities such as primary feasibility study, problem analysis, data collection, model design, model building and validation and finally implementation are normally carried out by individual consultants and different consultants have different methods of tackling the same simulation task. This may cause difficulties in future modifications, team integration and unreliable estimated project time. Therefore, he suggested the requirement of a more consistent approach and methodology to monitor the entire simulation process. During his research he has developed a common methodology supporting the process of building a simulation model. Techniques such as IDEF0, SSADM and flow charts have been applied to develop a methodology and it covers areas including overall project procedures, data collection form, project management guides, model building and result analysis methodology. However, this paper specially focuses on the overall simulation project management rather than input data modelling. It doesn't explain any logical data structure and data integration scheme for a database management system model.

Baum and Glassey(1992) adopted a structured data modelling approach which is used to maintain and support to simulation projects at NCR micro electronics product, USA. They have developed relational database management systems (RDBMS) to link their simulation tools and this RDBMS to store the information needed to define wafer fabrication operations at a level of detail required by the simulation application. The database consists of base tables, application specific tables, and tables for results. The base table contains the core data describing the manufacturing facility and process, application specific tables contain information which are particularly for the simulation and results tables to organise output data. This approach is only developed only for a specific industry; it does not cover all aspects of input data modelling.

It appears that most of the research undertaken to date has focused upon statistical data analysis techniques and less attention has been paid to developing methods to accelerate the modelling of input data needed in the simulation model. Due to the lack of literature in the area of data modelling, it has been necessary to look at data modelling methodology used in the other fields, such as Business Process Reengineering (BPR) and Enterprises Modelling (EM). However, it is not necessary to review the complete features of EM and BPR and it would be sufficient to concentrate on used methodologies and their applications which are relevant to the our study.

2.5 Data modelling in other areas

This section evaluates the data modelling approach in other areas such as business process reengineering (BPR) and enterprises modelling(EM). The reasons behind the selection of these specific areas, are that they have been applied very successfully and they are very popular fields today. Section 2.5.1 and 2.5.2 describe briefly how the data modelling methodologies are used in business process reengineering and enterprises modelling.

2.5.1 Business Process Reengineering (BPR)

Business Process Reengineering has become one of the most popular topics in organisational management creating new ways of doing business (Tunay, 1995). The definition of the BPR was given by Hammer and Champy (1993) as "the fundamental rethinking and radical redesign of business process to achieve dramatic improvements in critical contemporary measures of performance, such as cost, quality, service and speed". However, we will not review whole features of BPR and only will evaluate methodology which is mainly used for activity and data modelling.

Literature from the area of BPR projects found that one of the main project objectives is the development of a framework to capture business process changes and their process knowledge for redesign steps. It can be seen that most of the BPR projects have followed common methodology which is used to develop their BPR frameworks. More formally, a framework consists of two methods. One method has been used to capture or document the process of an organisation, i.e. to capture knowledge of how things work (AS-IS) or will work (TO-BE) in an organisation. This is called an "***activity modelling***". The other method has been used to capture what information is currently managed (AS-IS) or will be managed (TO-BE) in the organisation. This is called "***data modelling***" [Mayer et al (1995), Lejk and Deeks (1998), Kettinger et al (1995), Appleton, D S (1995), Huckvale and Ould Martyn (1995), Roberts (1997)].

Activity modelling. This is an important analysis part of almost every BPR project. Activity modelling is a technique which is used to understand how the business process really works. Activity modelling illustrates how things are happening in the system (called AS-IS modelling), and also how things need to be changed according to the redesign criteria (called TO-BE modelling).

Data modelling. Data modelling is used to describe how the information is shared by the different activities in terms of data relationships. Data models describe a current data structure for an "AS-IS" system as well as a new data structure for a "TO-BE" system. The hand book for business process improvement (prepared by the Federal Aviation Administration's Office of Information technology, 1995) highlights the definition of data modelling for the BPR.

"What is Data Modelling?

A data model is a fully-attributed business rule model, the purpose of which is to design or document a physical database. Data modelling is used to document the information requirements of the functional activities, to identify reuse requirements and opportunities, to assist in identification of redundant processes, and to guide consistent data administration."

Kettinger et al (1995) have mentioned many documentation techniques and tools that may be supported to document the process of business, namely, Data Flow Diagramming (DFD), Block Diagramming, Process Flow Charts and IDEF methodology. Hlupic (1998) has mentioned that a variety of software tools for BPR are available in the market and many of these tools provide graphical representations.

2.5.2 Enterprise Modelling (EM)

A definition of Enterprise modelling is given by Moynihan (1997) as "Enterprise modelling is the process which is to develop a repository regarding organisational elements and functions that maps information objectives with business functions".

Enterprises modelling provide a conceptual framework in terms of business function or activity, organisational information or data requirements which provides an integrated picture of the enterprise (Presley 1993, Moynihan 1997). A literature survey was conducted to find out existing enterprise modelling architectures. Most of the enterprise modelling architecture provide a framework for developing a functional view of an enterprise and function-information-dynamic representation of the existing operations of an enterprise with possible integration [(Jayaraman, (1990), Srinivasan & Jayaraman, (1997), Malhotra and Jayaraman, (1992), Cheng and Lu, (1996), Whiteman et al (1997), Moynihan (1997) and Vernadat (1993)]. Literature has established that function, information and dynamic model of an enterprises have been developed separately and later they have been integrated by using possible mapping techniques. A variety of methodologies have been developed to support this approach. However, several attempts have been made to integrate these three models.

Jayaraman has proposed a set of concepts for the selection of a methodology for modelling an enterprise (Jayaraman 1990). The methodology should;

1. Be able to express the various manufacturing operations of an enterprise in a natural language and in a straightforward way.
2. Permit a hierarchical decomposition, i.e., construct and view the desired level of detail of the system being modelled.
3. Be orientated towards well graphical representation.
4. Allow a wide range of users to communicate.
5. Be flexible and easy to use.
6. Be available in the public domain

It can be seen that most of the data or information models of Enterprise Modelling and Business Process modelling, described in the literature, have been developed to identify each and every item of an information and how information is shared by different functional areas. i.e. Design of an integrated information system for an organisation requires a complete understanding of the various functional activities involved in the system. The data or information model is concerned with "what" data or information should be involved in the system. The functional model is concerned with "what" is happening in the system or with explaining a system. Hence, it is necessary to review the some data modelling and the functional modelling techniques before developing our Reference Data model to identify and collect input data in the simulation of manufacturing system.

The literature survey established that, so far, there is no one methodology to describe the functional activities, dynamic behaviour and information characteristics of a system in one model. i.e. functional, dynamic and information modelling methodologies are separate and independent of each other [Singh et al (1996), Chen et al (1996), Wood et al (1986) and Wyatt et al (1990)]. One must integrate some different models together in order to analyse the system. Furthermore, the application of most currently available system designs and modelling techniques are primarily confined to the conceptual design phase, with a few of them able to extend support for the implementation phase. However, there are many separate and independent techniques available for functional modelling, dynamic modelling and data or information modelling of manufacturing systems. Two such systems are

- IDEF Methodology (Mayer and Painter, 1991)
- SSADM Methodology (Ashworth and Goodland, 1990)

The IDEF and SSADM methodologies were selected and will be reviewed in section 2.6 and 2.7 respectively. The IDEF and SSADM methods will also be evaluated in chapter 3 to select a suitable one for developing the proposed research methodology.

2.6 IDEF Methodology

The integrated Computer Aided Manufacturing (ICAM) project of the U.S. Air Force has developed the ICAM DEFinition (IDEF) methods to address the particular characteristics of manufacturing. IDEF methodology which may be applied to any manufacturing system that can be expressed as a characteristic of the system graphically. It can be used as support to answer the following three basic questions in order to understand the particular characteristics of a manufacturing system(Maji, 1988).

These are :-

- What functions are being performed ?
- What information and data is needed to support these functions ?
- What changes to the functions and information occur over a period of time?

IDEF methodology is comprised of three divisions which graphically characterise different aspects of the manufacturing system (Mayer, 1992). They can answer the above questions. The three divisions of IDEF methodologies are;

1. IDEF0 is used to produce a functional model ,
2. IDEF 1 is used to produce a information model and IDEF1X is used to produce a data model (IDEF1 Extended), and
3. IDEF2 is used to produce a dynamic model.

The IDEF family has now been extended from IDEF0 to IDEF6, including IDEF1X., namely, IDEF3- Process Flow and Object State Description Capture Method , IDEF4- Object-Oriented Design Method, and IDEF5- Ontology Description Capture Method, IDEF6-Design Rationale Capture Method. These techniques may be used independently. In 1993, the Computer System Laboratory of the US National Institute of Standards and Technology (NIST) released IDEF0 as a standard for Functional Modelling and IDEF1X as a standard for Data Modelling (Laamanen 1994)

2.6.1 IDEF0 Function Modelling Method

IDEF0 is a structured functional analysis technique for manufacturing. The IDEF0 model consists of a series of related diagrams organised in a hierarchical manner [Mayer, 1992(a)]. IDEF0 makes use of the hierarchical cell modelling graphical techniques to describe the functions at the desired level of detail. In a IDEF0 model, the central box represents the activity, described by an activity name beginning with a verb.

As shown in figure 2.1, arrows enter and exit the box. The arrows from the left represent input (I) to a activity and arrows coming out from the right represent the outputs (O) that the activity produces by transforming or consuming its inputs. The arrows coming from the top are controls (C) which constrain or control when or how the activity is accomplished. The mechanisms (M), the resources used to execute activity, enter from the bottom. The arrows in the activity model are known collectively as "ICOMs".

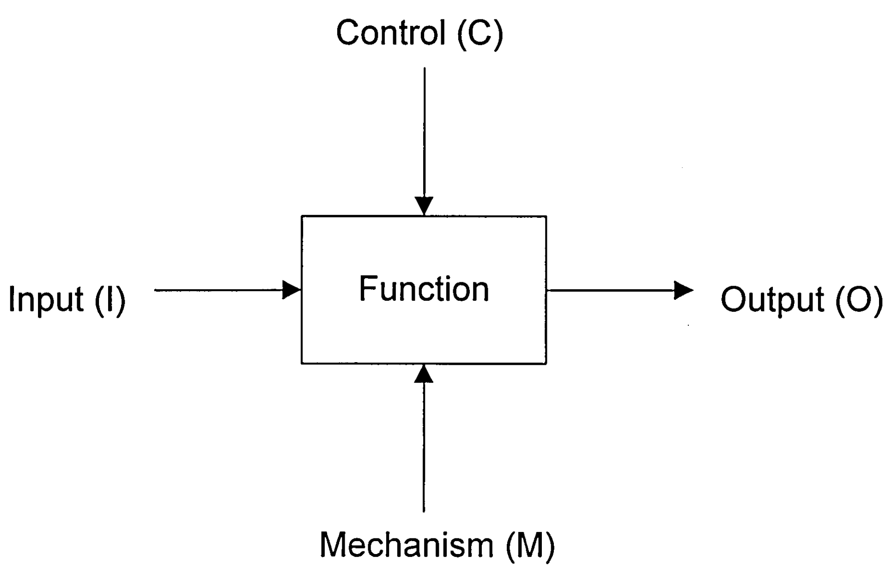


Figure 2.1 The ICOMs

Normally, the IDEF0 model starts from a general representation of the system. This representation is called A0 diagram. The decomposition process can be performed further, braking down the A0 diagram into sub-diagrams to describe as required the level of details as shown in figure 2.2.

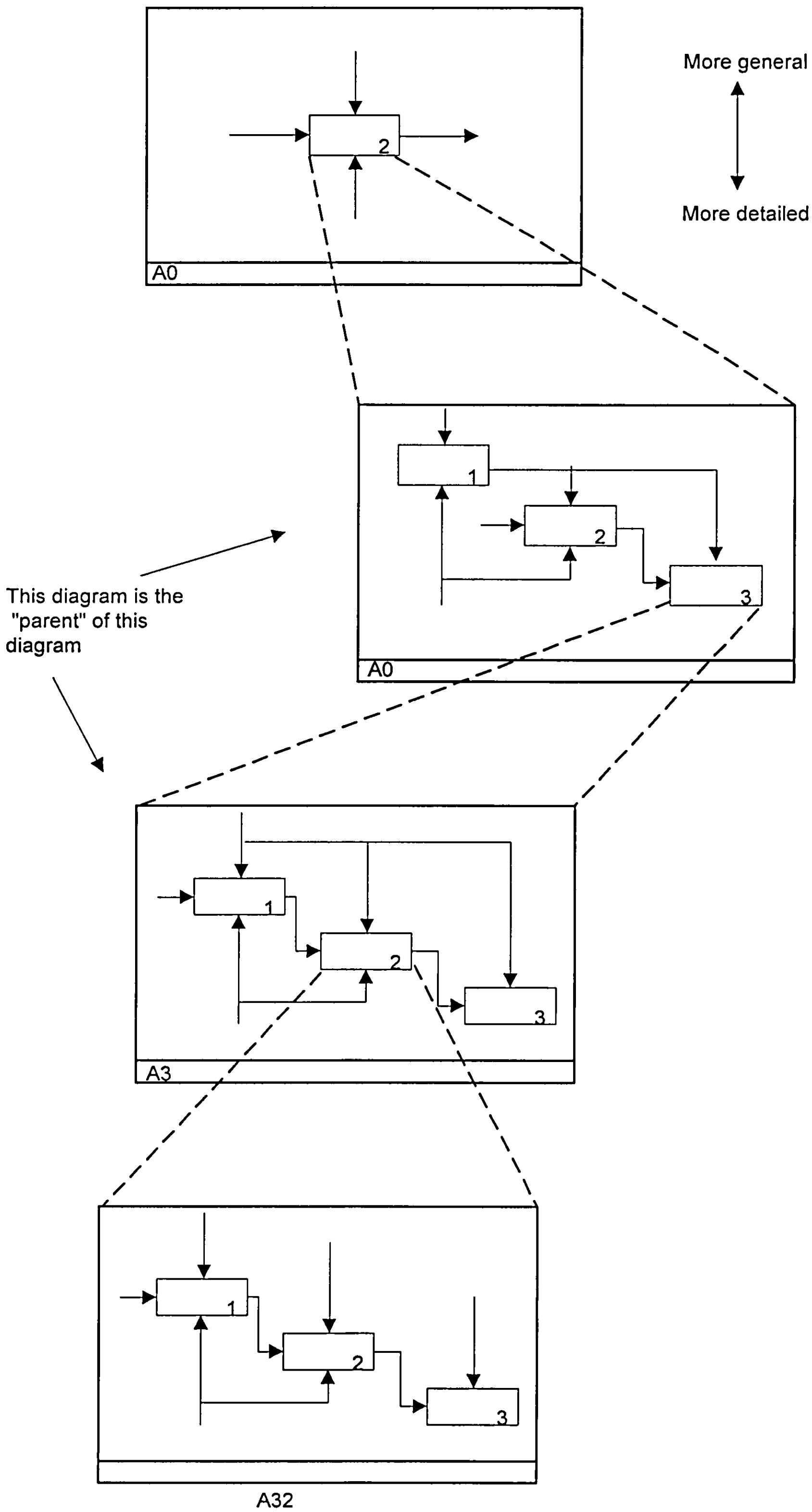


Figure 2.2. The decomposition of IDEF0 diagram

2.6.2 IDEF1X Data Modelling Method

IDEF1X is a method for designing relational databases with syntax designed to support the semantic constructs necessary in developing a conceptual implementation (Mayer 1992b). IDEF1X, especially addresses the logical structure of shared data, defining this structure in terms of entities, attributes of entities and relationships between entities. IDEF1X guides user terms throughout the development of data models, which lead to precise, normalised, graphical statements of meanings and structures. These data models represent the business rules of an organisation or system (Maji 1988).

2.6.3 IDEF 2 Dynamic Modelling Method

IDEF2 Dynamic Modelling Methodology is a tool for representing the time-varying behaviour of the functioning of a manufacturing system or environment (Mayer 1992a). To describe a system in IDEF2, the system should be divided into four sub models (Maji 1988 and Ralph et al 1985).

They are:

- Facility sub models: The facility sub models describe the resources which are used by the system to produce its output,
- Entity sub models: The entity sub-models graphically describes the flow of products and information through the facility,
- Resource disposition sub-model. The resource disposition model is used to describe the disposition of resources when they become available, and
- System control sub-model: The system control sub-model describes the occurrence of activities which control but do not prescribe the flow of entities. The situation handled by this sub-model includes the breakdown and repair of resources, the arrival of entities and the job priorities, etc.

IDEF2 was intended to be used as a dynamic modelling method for simulation. It is no longer referred to as IDEF2 because the market has accepted it as simulation (Whiteman et al , 1997). The SLAM simulation language is the commercialisation of the original IDEF2 and is based on the graphical notations of the IDEF2.

2.7 SSADM methodology

SSADM has been developed by Learmouth & Burchett Management Systems and the Central Computer and Telecommunication agency, UK. It has been used mainly in the field of designing commercial and administrative information systems since 1983.
The SSADM provides three divisions of diagrammatic techniques. They are

1. Data Flow Diagrams (DFD)
2. Logical Data Structure (LDS)
3. Entity Life History (ELH)

2.7.1 Data Flow Diagrams (DFD)

The data flow diagrams show the overall data flow through a system. Data flow diagrams are one of the most powerful and useful techniques available to the system analyst (Lejk and Deeks, 1998). Data flow diagrams show data stores and also external sources and the destination of data. They also show processes and the flow of data among those processes As in IDEF0, each process can be decomposed into lower levels.

2.7.2 Logical Data Structure (LDS)

Logical data structure is a technique used to model entities and their relationships in order to achieve a representation of the structure of data. LDS are concerned with modelling data. The ultimate purpose of logical data structuring is to create the basis for database design.

2.7.3 Entity Life Histories (ELH)

The entity life cycle is a technique used to diagrammatically represent how the system information changes over a period of its life time and the sequence of events which make the information change (Pandya et al , 1997 and Maji 1988).

2.8 Summary

The literature concluded that the modelling of input data for the simulation is one of the most important parts of the project and this task is usually most frustrating and time consuming. It proved that the data modelling is still an uncovered area in the simulation; specifically less attention has been paid to identify the problems that influence the input data modelling and development of guidelines to show how this task should be carried out.

The literature survey of this research attempted to find out causes of inefficient data modelling in simulation, but it can be seen that no attempt has been made to identify and properly document the spectrum of data modelling problems. It can be seen that only some issues regarding underlying factors which have direct impact on input data modelling exercise. However, these factors can influence the inefficient data modelling in hidden form, therefore, the details analysis of data modelling problems will be discussed in chapter 3. Some of these questions need to be answered to accelerate input data modelling exercise. Therefore, part of the literature survey was carried out to find current data modelling research in simulation. It appears that most of the research undertaken to date has focused upon statistical data analysis techniques and no or little work has been carried out to develop a method to accelerate data modelling exercises. Due to the lack of literature in the area of data modelling in simulation, the final part of the literature survey was carried out to find data modelling in other areas such as business process reengineering and enterprises modelling to select suitable data modelling techniques and methodologies to apply this research to develop a systematic approach to accelerate the input data modelling exercise.

CHAPTER 3

ANALYSIS OF INPUT DATA MODELLING PROBLEMS

3.1 Introduction

Prior to the development of a methodology for rapid input data modelling, it is necessary to identify the major problems which affect the modelling of input data. The literature survey concluded only the underlying factors of the problems relating to input data modelling, but no attempt has been made to identify a coherent view of the problems. Therefore, this chapter presents specific problems that have been synthesised on the basis of the underlying factors and the impact of such problems on the inefficient input data modelling in simulation. The chapter also describes the solution to the above problems together with the methodology that has been adopted to achieve the research objective.

The comprehensive evaluation of the IDEF and SSADM methodologies on the basis of their modelling features is also presented in this chapter in order to select a suitable one for developing the proposed research methodology.

3.2 Identification of Input Data Modelling Problems

The summarised literature is vague in describing the underlying factors in the problems relating to input data modelling in section 2.3. Having analysed these factors, all the potential problems associated with input data modelling have been listed under the key stages of a simulation life cycle as shown in table 3.1.

Key Stages	**Problems**
Conceptualisation	• Wrong problem definitions • Lack of clear objectives • System complexity • High level model details
Data Collection	• Poor data availability • Difficult in identifying available data sources
Model Construction	• Limited facilities in simulation software to store, organise and manipulate input data

Table 3.1 Potential problems associated with input data modelling

3.2.1 Wrong problem definitions

The first stage of any simulation project is the definition or formulation of a problem. Some situations in a manufacturing systems simulation project may involve poor or wrong definitions of problems. Shannon (1975) argues that in some simulation exercises, many millions of dollars are spent developing exotic solutions for the wrong problem. Simulation projects initiated with a poor understanding of problems have a higher risk of failure due to excessive time being invested in modelling inappropriate data (Tye and Perera, 1997).

Leung & Lai (1997) state that the correct problems can be identified through interviews, questionnaires and existing documents. Furthermore, they argue that in many cases the people who are responsible for problems in the domain area may not even be able to identify their actual problems. Leung & Lai (1997) also suggest that to identify and correct a valid problem, it is always beneficial to facilitate communication and problem identification through graphical tools.

3.2.2 Lack of clear objectives

One of the most important, but often neglected, parts of a simulation study is the definition of clear project objectives (Law 1990 and Robinsion 1994). All projects needs objectives. Yet, many projects start without clear objectives. This reason will have an enormous impact on the scope of the model, the level of details to be incorporated and hence, the data that needs to be collected (Novels, 1992). It is impossible to decide an appropriate level of model details and required data for the system without clear objectives and also needed to identify the modelling outputs to justify these objectives. This may cause the modeller to identify wrong, unnecessary model data which is not required for the model. Moreover, a significant amount of project time is spent on these model data. Well defined objectives are vital in determining what elements of the system need to be modelled and also what input data needs to be collected. Therefore, in simulation projects, the first step must always be to determine the clear objectives. The objective should contain more details to avoid using valuable time modelling unnecessary and incorrect data.

3.2.3 Complexity of the system under investigation

The variety and volume of data to be collected is very much dependant upon the complexity of the system under investigation. As the collation of data progresses, it is often necessary to cross-check data for its completeness and integrity. Where this on-going data validation and verification is not possible, several iterations are required before the appropriate and accurate database is established. Interviews with simulation practitioners revealed that data is often collected in ad-hoc fashion, particularly in the case of large and complex systems.

It has been noticed that even in very similar projects, different practitioners adopt different approaches to gather and analyse data. Even in a physically small manufacturing facility, depending upon the level of manufacturing activities, it may be necessary to gather a large variety of data. This reason will lead to reduced data handling capability. For example, the assembly planning activities data is very difficult to collect due to the increase of information about parts and their assembly attributes. Grewal et al (1995), have mentioned that in assembly operation, the number of tasks are always more than the number of parts, and the amount of information to be processed is dependent upon the number of tasks rather than the number of parts.

3.2.4 High level of model detail

The level of detail has clear implications on data modelling. However, one of the questions is how to determine the level of detail. The level of detail should depend on project objectives, data availability, creditability concerns, computer constraints and the opinion of system "experts" (Law, 1990). If the appropriate level of model detail is not determined at the outset, the gathering of required data becomes difficult. However, the maximum detail level and scope of the model must not lead to the higher accuracy and understandibility (Figure 3.1) but lead to longer data collection time (Figure 3.2).

For example, as shown in figure 3.3, a typical system component may have many attributes, out of which some can be core attributes and on which, every company may maintain records. Collection of this type of data would be less time consuming and would not require much effort. On the other hand, there can be some additional attributes that can be identified with an object but can not be measured directly or quantitatively. For instance, data items related to the machine such as set-up, breakdown data, efficiency and labour allocation require more effort and time to collect comparative to machine time. Table 3.2 shows the effort required to collect data related to the parts/materials, resources and rules based on the ranking system according to the results obtained from the analysis of a questionnaire survey conducted at 1997 Winter Simulation Conference (See Appendix B for the detail analysis of the questionnaire survey results).

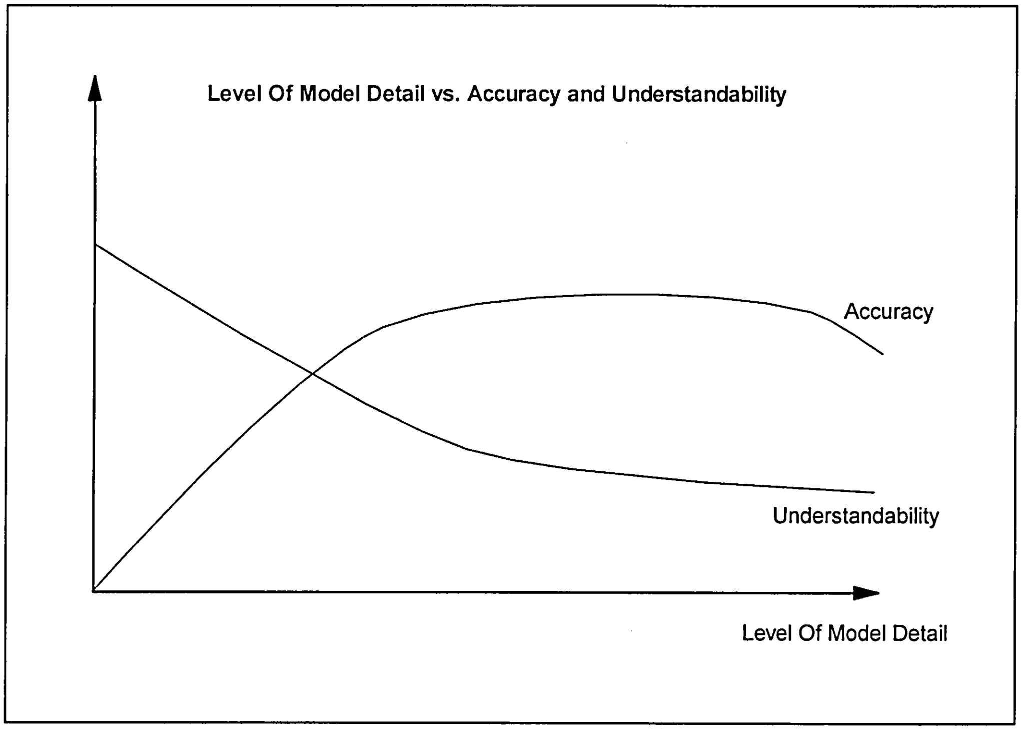


Figure 3.1- Level of model detail Vs Accuracy and Understandability (Robinsion, 1994)

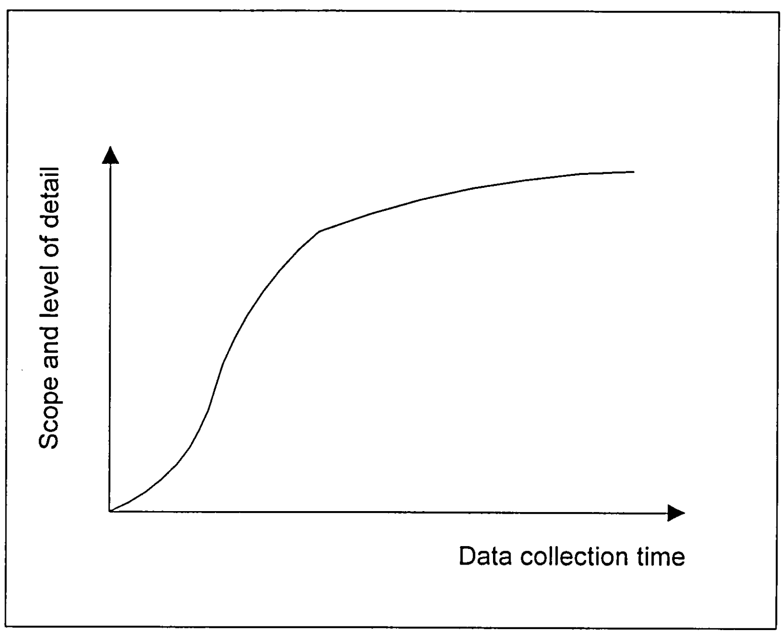


Figure 3.2 Data collection time Vs Scope and Level of model detail (Robinsion, 1994)

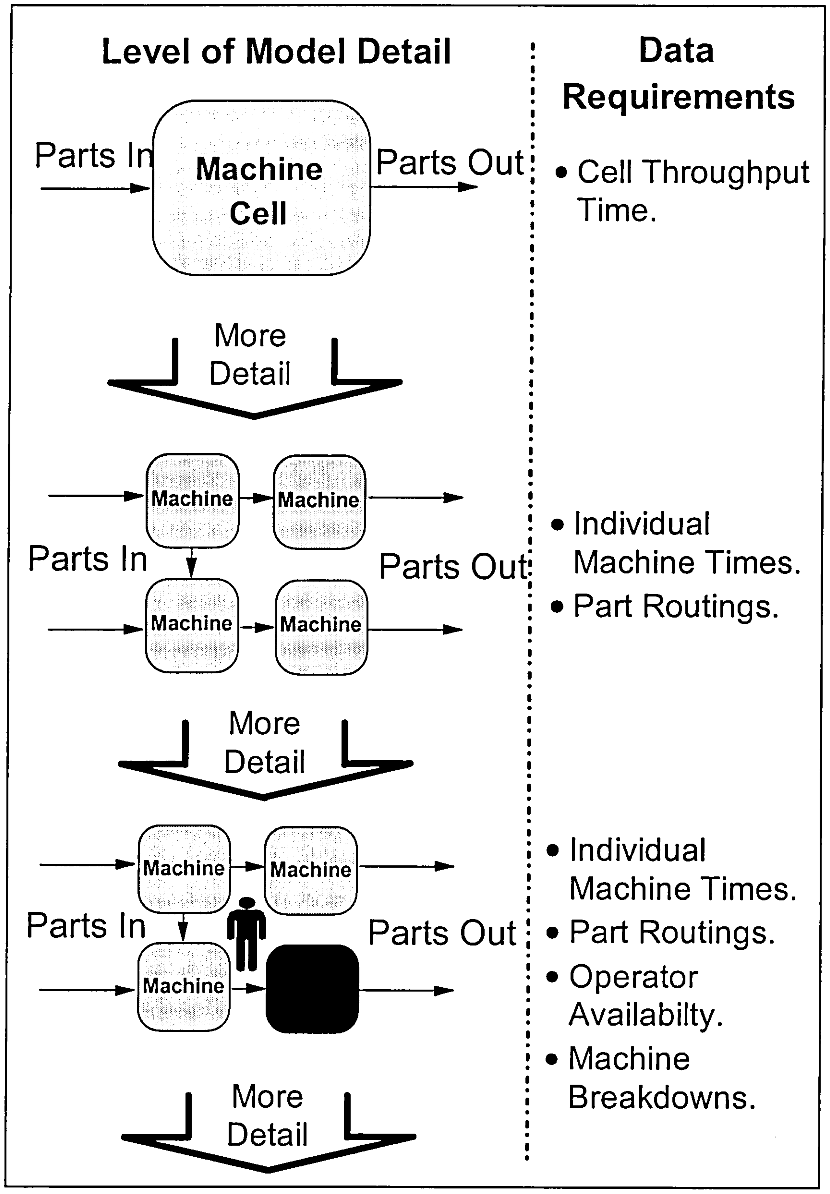


Figure 3.3.-Level of model detail Vs Data requirements (Tye 1997)

DATA ITEMS	RANK
Parts/Materials	
Arrivals time	1
Schedule data	2
BOM data	3
Resources	
Break down data	1
Efficiency	2
Labour allocation	3
Set-up time	4
Machine time	5
Logic	
Process routes	1
Priority	2

Table 3.2 Effort required to collect

The basic rule is to model the minimum amount of detail required to achieve the project objectives (Robinsion, 1994).

3.2.5 Poor data availability.

Two factors have been identified for poor data availability during the literature survey. When model details are increased, it may be difficult to find quality data for new attributes (Hatami 1990, Robinson 1994). The required data may simply not be available. Simulation models are also built for the systems which do not currently exist. In such cases, the modeller may not be able to collect the required data due to the unavailability of past operation data and no opportunity to collect them (Pegden et al 1990, Law et al 1994, Robinson and Bahatia 1995).

3.2.6 Difficulty in identifying available data sources

A simulation project is only as good as its data preparation. A sufficient amount of the accurate input data must be available in a required quantity and form within the company data sources. The simulation model needs substantial data from numerous data sources in the manufacturing company. Figure 3.4 summarises the main sources of data available from different sources. These data sources can vary from simple manual systems to sophisticated computer based systems. These systems can often provide the required data. Model builders, however, find it difficult to identify reliable source data due to;

- The existence of multiple data sources for the same data type. For example, processing time of parts may be found in both MRP II and process planning systems. Due to a lack of integration, these two sources may provide different values for similar types of data. This uncertainty in data may force the model builder to seek a third party opinion to identify a more accurate source of data.

- Indirect existence data. For instance, the data required to model machine breakdowns (mean time between failures and mean time for repairs) may not be directly available. The maintenance department or maintenance contractors may have required data in very crude form. Hence a considerable time is needed to collect and analyse this data.

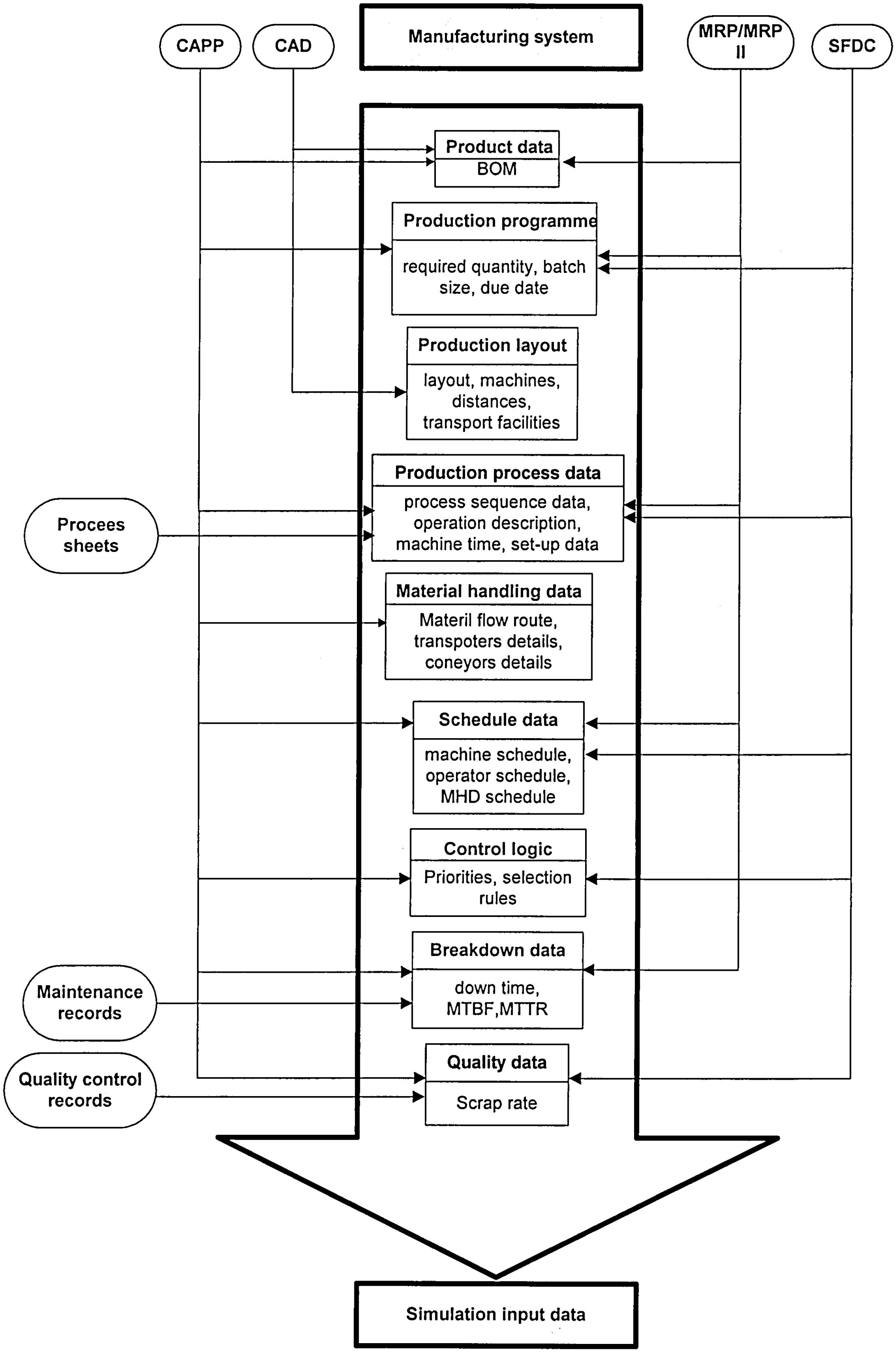


Figure 3.4 Data sources in manufacturing companies

3.2.7 Limited facilities in simulation software to organise and manipulate input data.

Many simulation software packages have been developed for the modelling of a manufacturing system but they have limited facilities to input data organisation and manipulation. They do not have proper database management systems facilities for collecting and documenting data. Frequently, the collated data is stored in propriety formats. When the core data is altered, it may be necessary to regenerate the data for simulation models. In large scale projects, this can be very time consuming. At present, the quality of input data handling of the simulation software is not high. Data can be entered into the model via a menu driven interface or can be read directly from the files. But it can be seen that at the moment, few simulators enable integration with spreadsheet packages and modern databases and it is not possible to integrate most of the packages with spreadsheets and databases.

Table 3.3 provides a comparison of a few manufacturing simulators with their input data handling capabilities (Hlupic 1993). The comparison is not considered in order to discover which is 'the best' simulator for input data handling because the main reason for this is a constant updating of existing software. At the moment, the quality of input data handling of software is not very high. However, it can be seen that all evaluated simulators enable integration with spreadsheet packages. ARENA ProModel and SIMFACTORY II.5 can be linked with databases and statistical packages respectively.

SIMULATOR	ARENA	ProModel	SIMFACTORY II.5	WITNESS	XCELL+
Quality of data storage, retrieval and manipulation facilities	Low	Low	Low	Low	Low
Model and data separation	✓	✓	✓	✓	X
Input data reading from files	✓	✓	✓	✓	✓
Rejection of illegal inputs	✓	X	✓	✓	
Integration with spreadsheet packages	✓	✓	✓	✓	✓
Integration with statistical packages	✓	✓	✓	X	X
Integration with CAD software	✓	X	X	X	X
Integration with DBMS	✓	X	✓	X	X
Integration with MRP Software	X	X	X	X	X
Integration with scheduling software	X	X	X	X	X

Table 3.3 Data handling capabilities of the simulation software (Adapted from Hlupic 1993) - (Key :- ✓ = Possible and X= Not Possible)

3.2.8 Impact of seven input data modelling problems

An analysis of all the problems can be considered as direct causes for excessive data modelling time, unrepresentative models and inaccurate results. The participants of the 1997 Winter Simulation Conference were asked to rank the impact of these problems on input data collection (see table 3.4). Poor data availability was considered a major reason for long data modelling time. (See Appendix B for the detail analysis of the questionnaire survey results).

Major reasons	Rank
Poor data availability	1
High level model details	2
Difficult in Identifying available data sources.	3
Complexity of the system under investigation.	4
Lack of clear objectives	5
Limited facilities in simulation software to organise and manipulate input data	6
Wrong problem definitions	7

Key: (1=most impact factor ,7= of minor impact factor)

Table 3.4 Impact of seven problems

Having analysed the input data modelling problems, an evaluation of the IDEF and SSADM methodologies on the basis of their modelling features will be discussed in section 3.3 in order to select a suitable one for developing the proposed research methodology and to address some of the data modelling problems.

3.3 Evaluation of IDEF and SSADM Methodologies

The literature survey investigated suitable graphical modelling methods used in the area of data modelling research. The, IDEF and SSADM methods are selected to evaluate the features to select the best one for our research. The features of these two methods will be evaluated in section 3.3.1. A comparison and conclusion of the comparison will be discussed in section 3.3.2 and 3.3.3 respectively.

3.3.1 Strengths and Weaknesses of IDEF and SSADM

Strengths of IDEF

1. IDEF0 activities can be described by their inputs, outputs, control and mechanisms (ICOMs).

2. Complex systems can be decomposed to whatever level of detail as desired, i.e., hierarchical approach.
3. The decomposition into lower level ensures the manageability of the model and more understandability of the complex system without losing the overall context.
4. IDEF0 diagrams illustrate the system that allows the human mind to deal with specific elements whilst retaining overall relationships.
5. IDEF0 models contain only few symbols; just arrows and boxes, therefore it is so easy to understand how the system works if the end user is a expert to the system or has only participated in the model development.
6. It has the potential to be used as an industry standard for manufacturing system design. (Wu, 1996)
7. IDEF1X can be easily translated into a relational database.
8. IDEF1X models are able to represent both dependent and independent entities separately.
9. IDEF1X allows categorisation of relationships to express entities within the system as mutually exclusive or inclusive.
10. Attributes can migrate automatically into an entity-in-model across relationships. This facility exists in some commercially available software.
11. Some commercially available software for IDEF1X can automatically generate SQL codes, or import SQL to reverse to reengineering existing databases into representative models.

On the other hand, IDEF is not without weakness. Below, we summaries some weaknesses of IDEF methodology.

Weaknesses of IDEF:

1. IDEF0 provides only a static representation of the process.
2. IDEF0 does not provide proper symbols for representing data stores and data sources.

3. The commercially available software packages for IDEF0 modelling are nothing more than a computer-aided drafting system. They have very little or no intelligence (Ang et al, 1997).
4. IDEF0 model does not denote the temporal relationships between the functions. Representation of temporal relations is essential for modelling and simulation of a manufacturing system.
5. The diagram does not make any distinction between data and material flow.
6. IDEF1X is only suitable for the relational database system.

Strengths of SSADM:

1. Possibility of defining what is happening in the "real world" of the day to day business.
2. DFD's are decomposed into desired levels of detail by separating each process into sub-process. i.e. hierarchical approach.
3. DFD's are a useful diagramming technique to illustrate the process , data store, external entity and data flow direction.
4. DFD are also a simple graphic technique, therefore, are easily understood by the end user and can be quickly modified.
5. LDS is a diagramming technique representing the entity or things in a system about which information is held and relationships or association between those entities.

Weaknesses of SSADM

1. When the amount of data to be handled becomes large, DFDs are difficult to create due to the existence of symbols such as data stores and external entity.
2. The use of a set of symbols (boxes, one-side open boxes, thin and arrows, ovals), contributes to distinguish the different entities, but can make the model messy, especially, as it has been pointed out, as soon as the complexity of the system increases a little (Pandya et al, 1997).

3. The commercially available software packages for SSADM are also nothing more than a computer-aided drafting system. They have very little or no intelligence.
4. As in IDEF methodology, a major short coming of SSADM is the lack of cohesion between the DFDs, LDS and ELH, i.e., integration of function, information and dynamic models into a single cohesive representation of the system.

3.3.2 Comparison of the IDEF and SSADM

This section is used to compare the features of two methodologies and the following categories are used to rank the methods .

- **Diagram Characteristics**

  Functional/Process Modelling

  This includes the ease of use of the diagrams and its syntax, flexibility of modelling features, representation of material & information flow and representation of data sources, external entities.

  Data Modelling

  The key features here are the implementation features of the relational database, flexibility of entity, attributes and relationship modelling capability.

- **Commercially available software and technical features**

  This includes a range of commercially available software to support these methods and technical features.

Features	IDEF	SSADM
	IDEF0	DFD
Diagram characterisation		
Ease of use	5	3
Input, Control, Output and Mechanism (ICOMs) representation	5	0
Hierarchical approach/functional decomposition	5	4
Interactive graphical capability	5	3
Static modelling features	4	4
Dynamic modelling features	0	0
Temporal relationships representation	0	0
Material flow	4	3
Information flow	2	5
Data & External sources representation	0	5
Commercial and technical features		
Range of commercially available packages	5	3
Integration between functional and information views	0	0
	IDEF1X	LDS
Diagram characterisation		
Relational database implementation	5	4
Separation of dependent & independent entities	5	0
Catergerization relationships	5	0
Commercial and technical features		
Capability of automatic migration of attributes	5	0
Generation SQL code	4	0

key (5= good, 3- average , 1=weak, 0-not provided)

Table 3.5 Comparison Table

3.3.3 Conclusion of Comparison

The basic idea of IDEF0 and DFD are very similar. From the above comparison (table 3.5), it can be proved that both the IDEF and SSADM methodologies are suitable for manufacturing system design and analysis. However, no methods are available to provide an integration approach between functional, dynamic and information model. Similarly, both IDEF and SSADM families provide separate and independent models for function , information and dynamic and there is no way to integrate models into a single cohesive system. Most of the authors argue that IDEF is a more appropriate methodology for the manufacturing environment (Ang et al, 1997 and Wu, 1996) and that SSADM came from system engineering. They also argue that IDEF was specially developed for manufacturing. Pandya (1994) and Wu (1994) have mentioned that of all the currently available graphical modelling tools for system description, none can be said to be the best for all purposes. This is an accurate statement. When choosing the tool for a particular modelling environment, there are a mixture of advantages as well as disadvantages.

However, the authors' personal opinion is that DFDs are suited for functional modelling when the sources/destinations of data are needed to be considered. Otherwise, IDEF0 is well suited for functional modelling and IDEF1X is a most powerful technique for data modelling.

3.4 Proposed Research Methodology

Based on the outcomes of the literature survey, presented in chapter 2, and an analysis of input data collection problems, presented in section 3.2, the author identified seven major problems which lead to longer data modelling time.

After analysing data modelling problems in section 3.2, it is necessary to address some of the major identified questions to accelerate the input data modelling exercises. It is obvious that a more methodical approach is required to identify and collate required data. Hence, there is a need to investigate a suitable methodical approach to tackle these questions and give guidelines to overcome these problems as much as we can. This section describes the initial background to tackle these problems and select the research methodology to model the development of this study. Figure 3.5 summarises the process of selection of research methodology to tackle some of these questions.

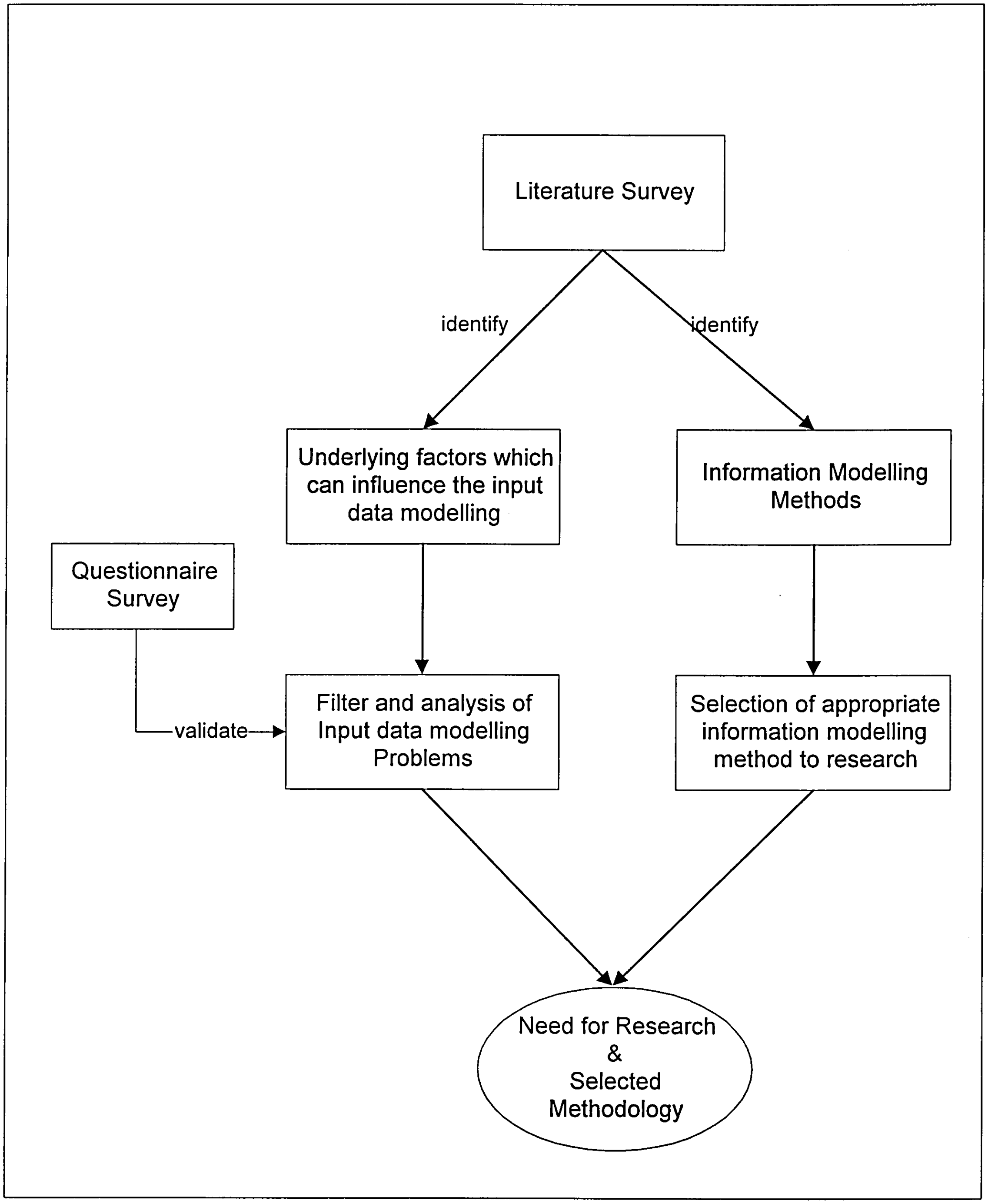


Figure 3.5 Selection of Proposed Research Methodology

Simulation modelling requires large amounts of data about the system being modelled. There is a need for the development of a generic reference data model to describe input data requirement for batch manufacturing projects. This can be used as a source of reference for rapid identification and the collection of data required for a large complex model according to the various level of detail and complexity. Such a fully-attributed reference data model can identify data entity (data classes), their attributes (data elements) and also capture entity relationships and cardinality.

Automated data modelling is a key component of the modern Computer-Aided Software/System Engineering (CASE) tool (Douglas,1997). These automated data models generate the entire database structure, along with its implied rules and relationships. Therefore, the development of a database management system (DBMS) can provide proper data handling capabilities for the simulation environment and also some kind of sequence of collecting data in a hierarchical manner.

According to the above discussion, it is obvious that such a methodical approach can contribute to tackle these problems :-

- Limited data handling capability in simulation software by handling data more effectively and efficiency within the DBMS.
- High model complexity by collecting data in a pre-defined sequence through a series of tasks to ensure complete and efficient data collection.

However, it is obvious that such a complex generic reference data model describes almost all data requirements for the simulation of a generic batch manufacturing system, but all the content data may not be needed for a given particular simulation project.

Furthermore, specific data may be unique to a particular manufacturing domain to achieve project objectives. There is a need to build a customised data model (entity model) for given systems under investigation. The development of such a customised data model for a given batch manufacturing system requires a complete understanding of the various phases of a simulation project such as system investigation, problem & objective definitions and the level of model details definition. It is clear that to cope with system investigation, problem formulation, objective definition and definition of the level of detail, we need to understand "how things work" in terms of a simulation point of view, i.e. the system should be described in detail (how the system operates). Therefore, it is necessary to map the process needed to construct simulation model and data needed to perform activities of those process (A process is thus a sequence of activities). Hence, the development of a customised data model (entity model) for a given system requires a complete understanding of the various activities of a system which are often used in simulation. It is obvious that activity models are considered as a static model and they are unable to capture the dynamic behaviour of the system (simulation is dynamic modelling). However, static models attempt to provide a static representation of dynamic systems (whitman et al, 1997). Static models generally explain the flow path of an object through a system. This information is helpful in determining what elements are involved in the activities performed by the system.

It is evident from the above discussion that an integrated approach is required to identify and collect data for a given system. Perhaps, the way is to develop activity and data models into a single integrated framework since activity and data are closely inter-related to each other. However, the literature survey has established that there is no method to describe activities and data within a single system. Due to the lack of cohesion between activity and data models, our research sought to design and develop a integrated **M**ethodology for **M**odelling of **D**ata (MMOD) in terms of activity and data views of a system for a real batch manufacturing simulation environment.

The activity diagrams used in the MMOD approach shows the decomposition of the generic manufacturing activities needed to construct simulation models. The static aspect of these activities describe their involving inputs, outputs, control and resources or mechanism. The data needed to perform these activities are described by the matched data reference model. The proposed methodology assists the modeller to generate the required data model (entity models) for a given batch manufacturing system under investigation and the development of a proposed methodology and use of the methodology describe in chapters 4 and 5 respectively.

The proposed methodology can also tackle other identified problems. The activity models describes the structure and behaviour of a system in terms of input, output, controls and mechanisms, Hence, the activity models help in structuring ideas about the problems and objectives to be tackled. On the basis of the level of details, the proposed methodology intends to identify data requirements. Table 3.6 briefly explains, how to use this methodology to tackle these problems in terms of information modelling.

Problem	Suggestions
Wrong problem definitions Lack of clear objectives	By developing activity modelling diagrams in simulation point of view, can support the problem formulation and objective definition phases • Identify key problems to discuss and data collection. • Enforce step-by-step approach. • Understanding of the "AS-IS" and "TO-BE" environment. • Encourage the modeller to think.
System complexity	By developing activity and data models • Help to understand large, complex systems • Both models can provide a method for capturing, organising and documenting the information about a complex system. • Combining both activity and data models, it is possible for modeller to identify data requirements according to the level of detail. • Identify & collect data in sequence manner. • Development of a proper database management system (DBMS) to organise and manipulate the large volume of data.
High level of model detail	By developing activity and data models • Activity models can be decomposed in hierarchical nature and modeller can easily start with a higher-level view to lower-level view by increasing level of details he or she desires. • Combining both activity and data models, it is possible for modeller to identify data requirements according to the level of detail. • By screening core data and optional data within the DBMS
Poor data availability	There is no easy solution to this question
Difficult in identifying available data sources	By developing a data source matched data model • Link data sources with reference data model
Limited data handling facilities in simulation software	By developing data model & DBMS • Development of a Reference data model and then translation to proper DBMS to store collected data. • DBMS is used to maintain the input data separate form simulation software, and then directly linked in to simulation models

Table 3.6 Suggestions to tackle the problems

CHAPTER 4

DESIGN AND DEVELOPMENT OF A METHODOLOGY FOR MODELLING OF DATA (MMOD)

4.1 Introduction

Literature on input data modelling in simulation concluded that a more methodical approach is required to identify, collect, organise and validate input data. Perhaps, the best way is to develop a guided system which could lead model builders through a series of tasks to ensure complete and efficient data identification, collection, organisation, manipulation and validation through the simulation life cycle.

The essential requirements of such a guided system were discussed in chapter 3. It explained that there is a need for the development of a generic framework in terms of batch manufacturing activities, which are often modelled in simulation, and data needed to support these activities. Hence, the architecture for such a generic framework should be consisted of an activity model and a data model. However, the development of IDEF activity and data models provide two distinct and separate views within a single framework; they do not provide an integration between models. Therefore, as a part of a framework, the mapping tables were developed to maintain the integration between models. This objective is achieved by linking all the activity components to appropriate data groups in the reference data model via mapping tables. The design and development of a generic framework is called MMOD (Methodology for Modelling of Data) and is described in this chapter.

4.2 Architecture of MMOD

The development of a MMOD approach for a generic batch manufacturing simulation project consists of three integrated phases as shown in figure 4.1.

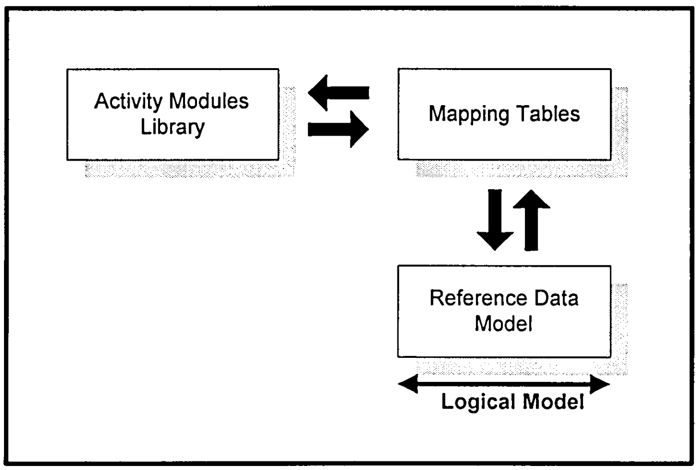


Figure 4.1 The architecture of MMOD

- **Phase 1: Activity module (IDEF0) library**

The activity model is a representation of the various processes, with their chronological sequence of activities in manufacturing systems as are often used in simulation. Having analysed the features of batch manufacturing systems, a series of activity models are documented in the library. The library contains the generic diagrams concerning machine operation, inspection, material handling and storage operations. The proposed activity modules library for the MMOD will be discussed in section 4.3.

- **Phase 2: Reference data (IDEF1X) model**

The purpose of a reference data model is to describe data integration between various components, such as parts, resources and system logic of a manufacturing system, into a single cohesive system and to describe conceptual database implementation. The reference data model specifically describes the structure of the data that is needed to perform activities which are often needed to build simulation models. These activities have been documented in the activity module library. The resulting data model (entity model) can be converted into a physical database (i.e. practical application of a data model), which shows exactly how the entities and relationships will be transformed into an actual database implementation. The development of a comprehensive reference data model will be described in section 4.4.

- **Phase 3: Mapping tables**

The purpose of mapping tables are to integrate both activity modelling diagrams and appropriate reference model data groups within a single framework so that the modeller can identify system activities, corresponding information and data quickly, to generate customise data models using pre-build IDEF models. The mapping tables also map the activities that the reference model data, with corresponding ARENA simulation software program constructs. The mapping tables used to integrate both activity and data modes are presented in section 4.5.

The key idea underlying MMOD is to generate a customised data model (entity model) for a given simulation project as explained in section 3.4 The steps for the development of a customised entity model, through the simulation project life cycle, will be discussed in chapter 5.

4.3 PHASE 1: ACTIVITY MODEL LIBRARY

4.3.1 Overview

The main purpose of activity modelling diagrams is to show clearly, the basic structural behaviour of system resources (machine, people, transporters etc.) and their relationships with entities (parts, material, etc.) from a simulation point of view. Harrel and Field (1996) argue that the concept of integrating process mapping with simulation technology is not new. Several attempts have been made to integrate process with a commercially available simulation product. However, he points out that no method has successfully been applied to integrate process mapping with simulation technology, due to the lack of process modelling capability to describe simulation complexity. In general, process models as considered as static models and therefore, unable to capture the whole dynamic behaviour of a simulation model. On the other hand, static models attempt to provide a static representation of a dynamic system and identify the vast majority of information needed to construct a simulation model (Whitman et al, 1997). It is evident that these diagrams can also provide an environment for the simulation modeller to develop conceptual modelling diagrams at various stages of the model design phase.

There are many methodologies available for activity modelling including IDEF activity modelling methodology, as described in section 2.5. After an evaluation in section 3.3, the IDEF0 activity modelling methodology was selected to develop an activity modules library for the MMOD.

4.3.2 IDEF0 activity modelling

IDEF0 models are extremely useful for the static modelling of manufacturing systems. They have enabled the description of manufacturing activities in terms of their input, outputs, controls and the mechanism or resources (ICOMs). This arrangement facilitates the definition of parts, resources and control rules which are used in simulation. This information is therefore very useful in determining what elements are involved in the activities performed by the system. Some of the other strengths of IDEF0 such as; simplicity and ease of understanding, hierarchical modelling capability, etc., make IDEF0 a suitable methodology for activity modelling. The strengths of IDEF0 methodology for activity modelling were discussed in chapter 3.

4.3.3 Identifying batch manufacturing activities

For an activity model library, the activities of the system being modelled need to be described and properly documented. Before the development of activity modelling diagrams for a library, it is necessary to understand and document how batch manufacturing activities operate and how it relates to other manufacturing activities in terms of a simulation point view. However, it is very difficult to analysis activities for a generic batch manufacturing system without the use of an actual model. The main reason is that many batch manufacturing systems are too complex . The interactions between the various system components in system may be very difficult to predict, especially in a generic manufacturing system. The complexity of the activity structure of generic batch manufacturing systems tends to confuse human comprehension. Therefore, to understand this complexity, it is necessary to structure and document batch manufacturing activities in general terms. Section 4.3.4 attempts to classify the generic batch manufacturing system components and activities; they will greatly aid the activity model library development stage.

4.3.4 Generic batch manufacturing system components and activities

This part of the research specifically focuses on the need for an analysis of generic manufacturing activities. This section will therefore explain a number of reasons and assumptions which are used to understand the basic structure of activities in the manufacturing system. However, the activities of manufacturing systems are much too complex to be represented and illustrated in the modelling diagrams. This is because of their vast amount of system components and shop floor activities. To reduce this extreme complexity, there are many views and ways of looking at the overall system. To do this, we can categorise system components and their activities. This enables us to describe the overall behaviour of a system.

Clark (1995) explains that manufacturing system components can be people, machines, tools, material handling devices, and materials. Meanwhile, Engelike et al (1985) have mentioned that, in general, shop floor activities can be divided into four classes, viz., MAKE, VERIFY, MOVE, REST. He explained these four activities as follows. MAKE is the manufacturing of parts, and for a specific industry, manufacturing parts includes the combination of several processes like milling, drilling, grinding, etc. The activity MAKE also includes assembly and disassembly activities. VERIFY is any kind of manufacturing activity in terms of inspection or testing. Generally, it can be any inspection. MOVE is a change of position or location. Basically, it can be transport , handling of parts or any supporting activity for the change of location such as loading, unloading. REST is planned storing (storage) or unplanned waiting (queue) before or after the MOVE, MAKE or VERIFY activities (see fig. 4.3).

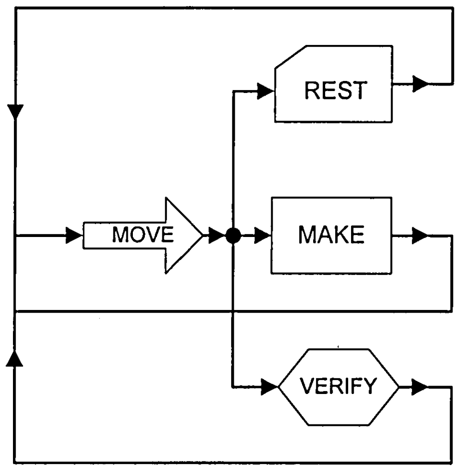


Figure 4.3 Basic Batch manufacturing operations

There are many classification schemes for manufacturing systems. However, some of them are not useful for our research. According to Clark (1995), Engelike et al (1985) and Ang et at (1997), with regard to the more generic manufacturing system, system components can be people, machines, tools , material handling devices, parts and storage facilities. More generic shop floor activities can be make, move, verify and rest (wait or store). For the purposes of this research, it is not necessary to consider tooling data. Therefore, tools are neglected just for this research. Unplanned waiting means queue. This relates to the simulation output result for queue data and is not considered in this research. Thus system components are used for generic manufacturing systems such as people, machines, material handling devices, parts and storage facilities (see figure 4.4) and genetic shop floor activities such as process (for assembly and non-assembly), inspection, handling and storage (see figure 4.5).

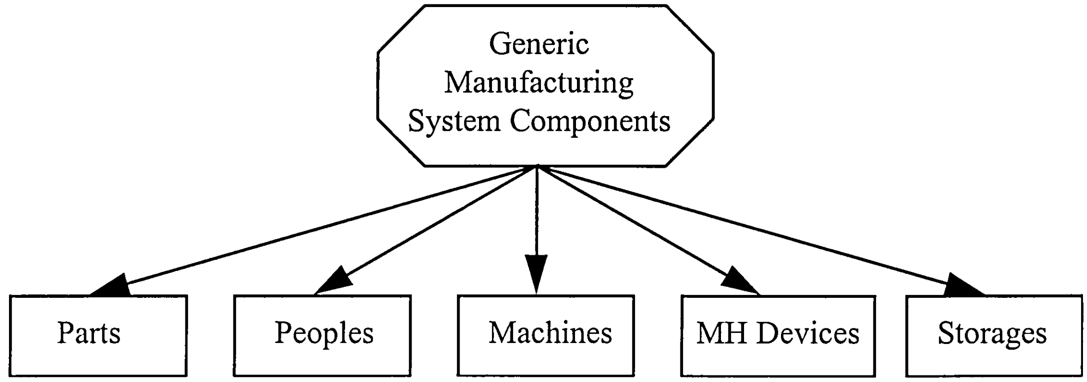


Figure 4.4 System components in batch manufacturing system

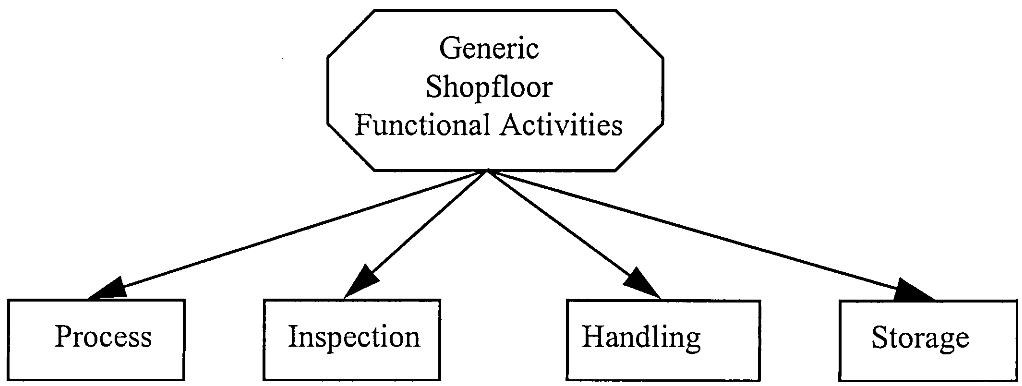


Figure 4.5 Generic slopfloor activities

According to the above classification for an activity modelling diagram library, there are main four generic activities (Process, Inspection, Handling and Storage operations). They will have been modelled with their associate hierarchies with necessary ICOMs.

For the activity model, the source activity represents one of the shop floor activities (i.e. Process, Inspection, Handling and Storage). Input (I) for the activity represents the parts. Mechanism (M) represents the machine, people or material handling devices. Finally control (C) represent all selection rules, priority data and scheduling (see table 4.1).

Shopfloor Activities	Input (I)	Mechanism (M)	Control (C)	Output (O)
Process for assembly and non-assembly	Parts	People Machines	Instructions Scheduling	Processed parts
Inspection	Parts	People Inspect Machines	Instructions Scheduling	Inspected Parts
Handling	Parts	People MH Devices	Instructions Scheduling	Parts Movement
Storage Operation	Parts	Storage facility	Capacity	Stored Parts

Table 4.1. Mapping between system components/ shop floor activities Vs ICOMs

The first step in developing activity diagrams is to characterise the major four shopfloor activities in terms of their necessary inputs, controls, mechanisms, and outputs, with associate decomposition diagrams which are expanded to the required level of detail. The highest-level activity block describes the main purpose of the activity and the lower-level activity blocks describes the supporting sub-systems which exist to serve the upper levels.

For a simulation project, activities (process, inspection, handling and storage operation) are modelled by any combination of three types of influences, viz. input, mechanism and controls. Inputs represent the flow of entity (single part or material and parts to be assembled) required by the activity. Control represents the instruction and schedule for the activity to perform. Mechanism represents the resources (machine, operators, MH devices, etc.) required by the activity.

4.3.5 Activity Modelling Diagrams Library

This section describes activity models and assists us in understanding manufacturing activities and how they operate in terms of a simulation point of view. This information will assist to identify and define entities, attributes and relationships which are used to develop a reference data model.

There are many user friendly computer aided software (computer aided system engineering)-(CASE) tools are commercially available to support IDEF0 structured activity modelling. An AI0 WIN (KBSI, 1993) activity modelling tool is used in this research to produce a activity modelling library which documents manufacturing activities, their relationships, and their associated inputs, controls, outputs, and mechanisms.

We have developed four standard IDEF0 activity modules for generic shop floor activities. These are identified in section 4.3.5. The modules are described as four major shop floor activities with associated decomposition.

- Module 1: Machine Process for
    - ⇨ Non-assembly
    - ⇨ Assembly
- Module 2: Inspection
- Module 3: Material Handling
- Module 4: Storage Operation

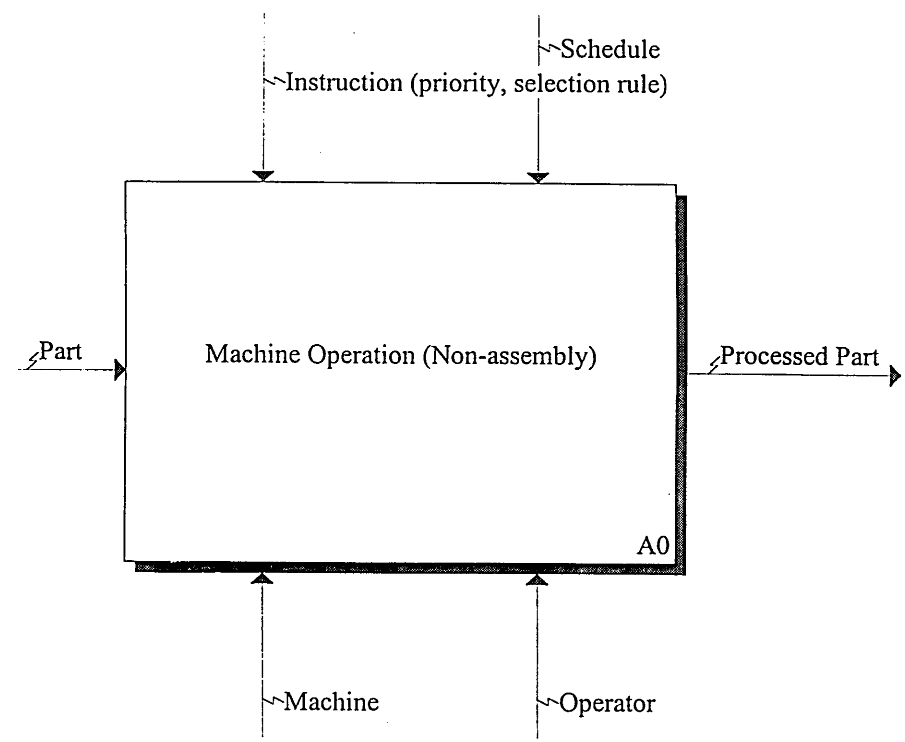
Module 1

Machine Process (None-assembly and Assembly)

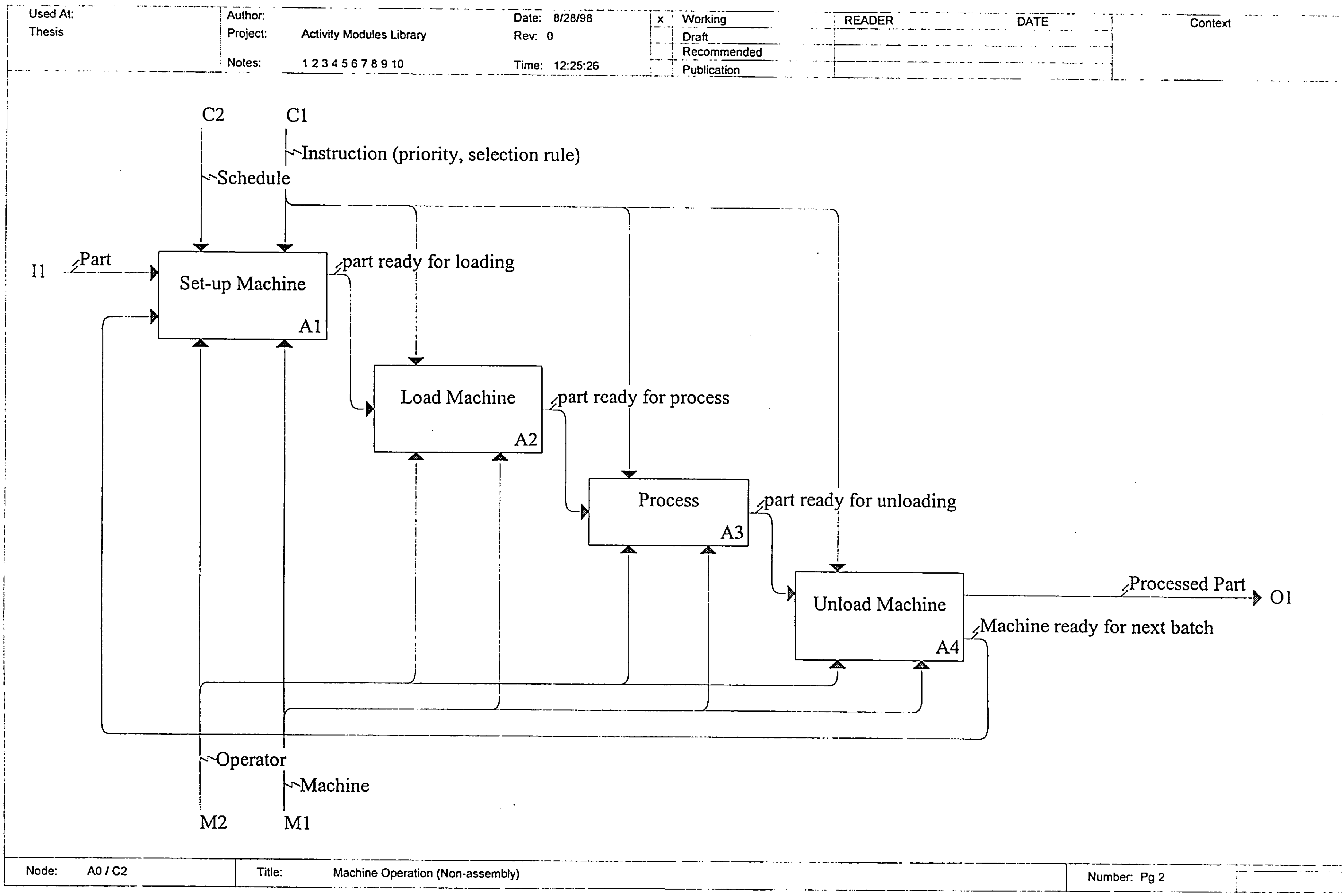
Machine activities concerning parts are common in almost every manufacturing simulation project. This diagram describes required machine activities concerning part processing. The initial diagram of part processing is general and indicates the activity "Machine Operation". This general diagram is said to be the parent of the details machine activity diagram. In the machine activity diagram, A0 diagram (context diagram) is decomposed into four sub activities as shown in diagram.

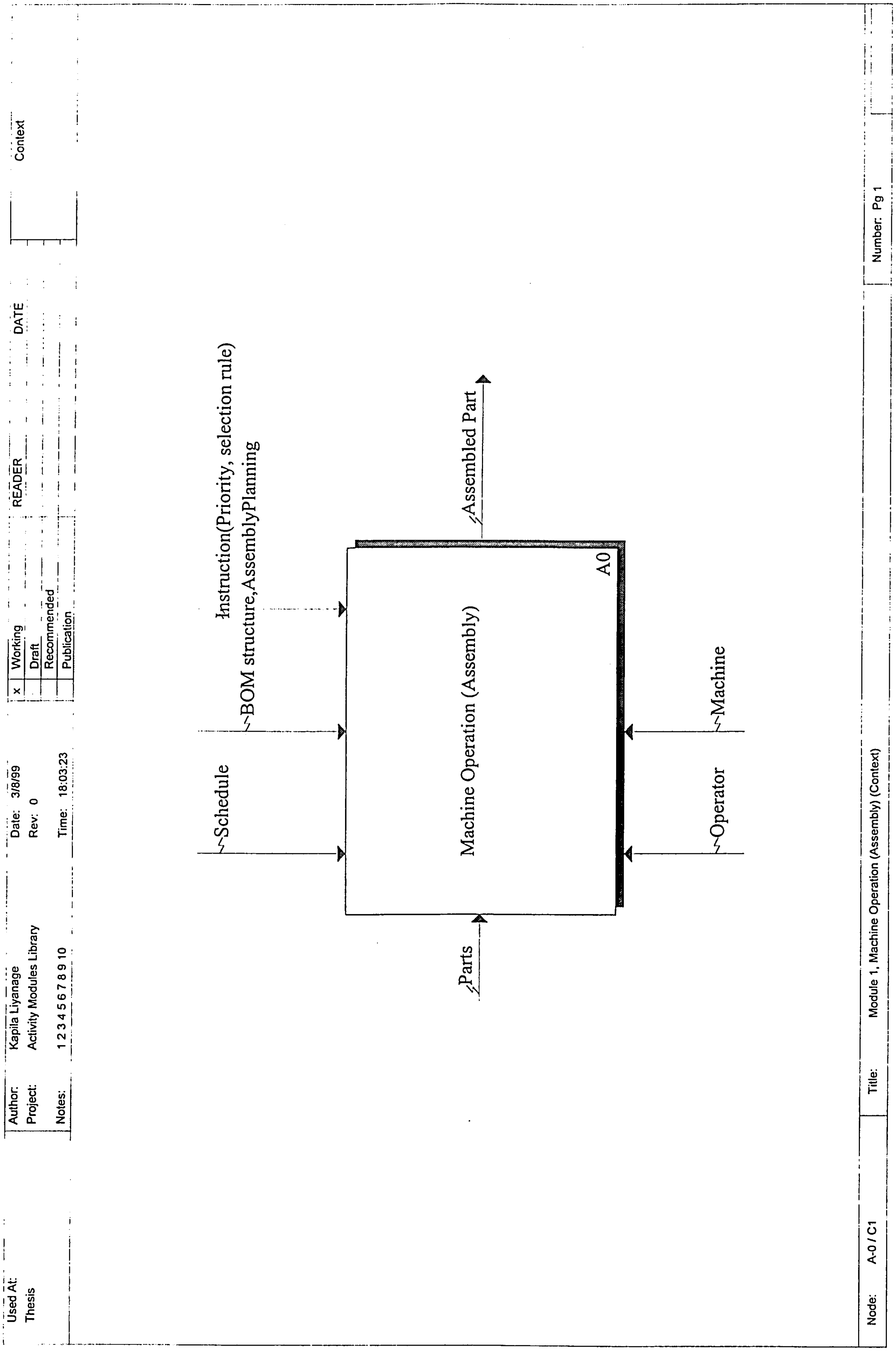
Used At: Thesis	Author: Kapila Liyanage Project: Activity Modules Library Notes: 1 2 3 4 5 6 7 8 9 10	Date: 3/8/99 Rev: 0 Time: 18:02:13	x Working Draft Recommended Publication

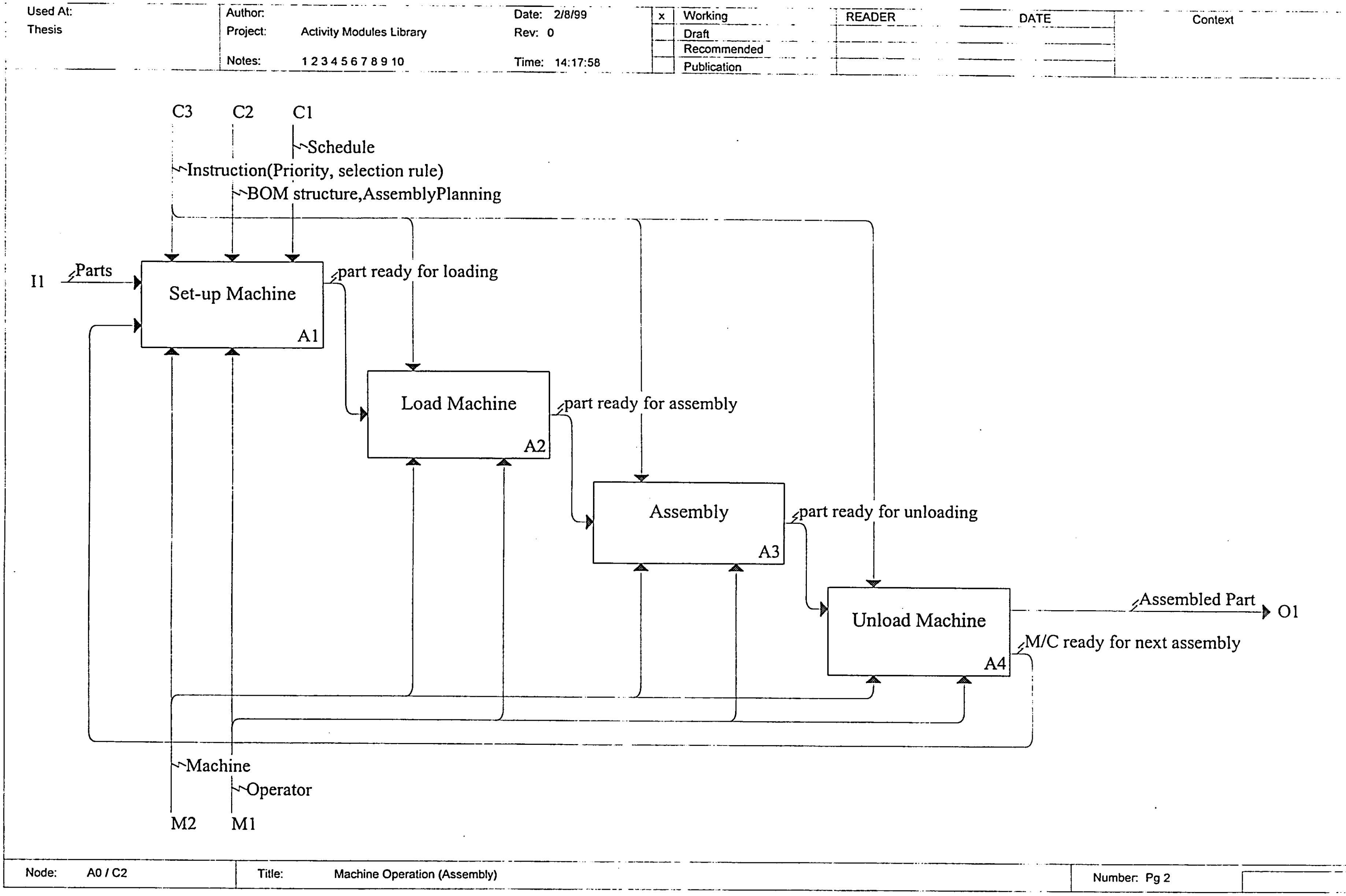
READER	DATE	Context



Node: A-0 / C1 | Title: Module 1, Machine Operation (Non-assembly) (Context) | Number: Pg 1


Used At:
Thesis
Author:
Project: Activity Modules Library
Notes: 1 2 3 4 5 6 7 8 9 10
Date: 8/28/98
Rev: 0
Time: 12:25:26
x Working
Draft
Recommended
Publication
READER
DATE
Context
C2
C1
Instruction (priority, selection rule)
Schedule
I1
Part
Set-up Machine
A1
part ready for loading
Load Machine
A2
part ready for process
Process
A3
part ready for unloading
Unload Machine
A4
Processed Part
O1
Machine ready for next batch
Operator
Machine
M2
M1
Node: A0 / C2
Title: Machine Operation (Non-assembly)
Number: Pg 2


Used At: Thesis
Author: Kapila Liyanage
Project: Activity Modules Library
Notes: 1 2 3 4 5 6 7 8 9 10
Date: 3/8/99
Rev: 0
Time: 18:03:23
x Working
Draft
Recommended
Publication
READER
DATE
Context
Schedule
BOM structure,AssemblyPlanning
Instruction(Priority, selection rule)
Assembled Part
Machine Operation (Assembly)
A0
Parts
Operator
Machine
Node: A-0 / C1
Title: Module 1, Machine Operation (Assembly) (Context)
Number: Pg 1


Used At:
Thesis
Author:
Project: Activity Modules Library
Notes: 1 2 3 4 5 6 7 8 9 10
Date: 2/8/99
Rev: 0
Time: 14:17:58
x Working
Draft
Recommended
Publication
READER
DATE
Context
C3
C2
C1
Schedule
Instruction(Priority, selection rule)
BOM structure,AssemblyPlanning
I1
Parts
Set-up Machine
A1
part ready for loading
Load Machine
A2
part ready for assembly
Assembly
A3
part ready for unloading
Unload Machine
A4
Assembled Part
O1
M/C ready for next assembly
Machine
Operator
M2
M1
Node: A0 / C2
Title: Machine Operation (Assembly)
Number: Pg 2

Module 2

Inspection

Activity inspection is similar to normal machine process except that the inspection indicates the fraction of part that passes or fails inspection, and allows for the rerouting to different destinations and the scrapping of parts.

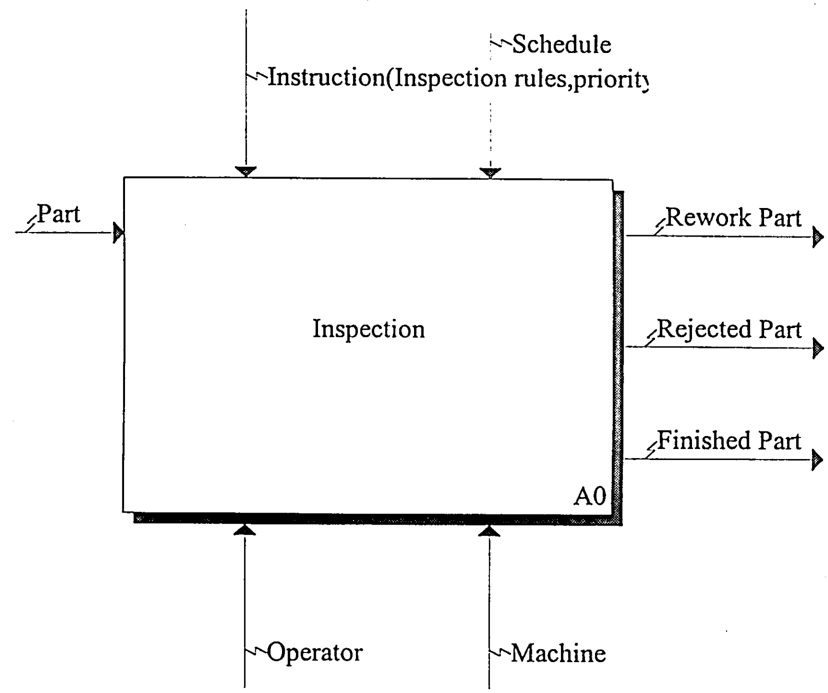
Used At: Thesis

Author: Kapila Liyanage
Project: Activity Modules Library
Notes: 1 2 3 4 5 6 7 8 9 10

Date: 3/8/99
Rev: 0
Time: 18:04:13

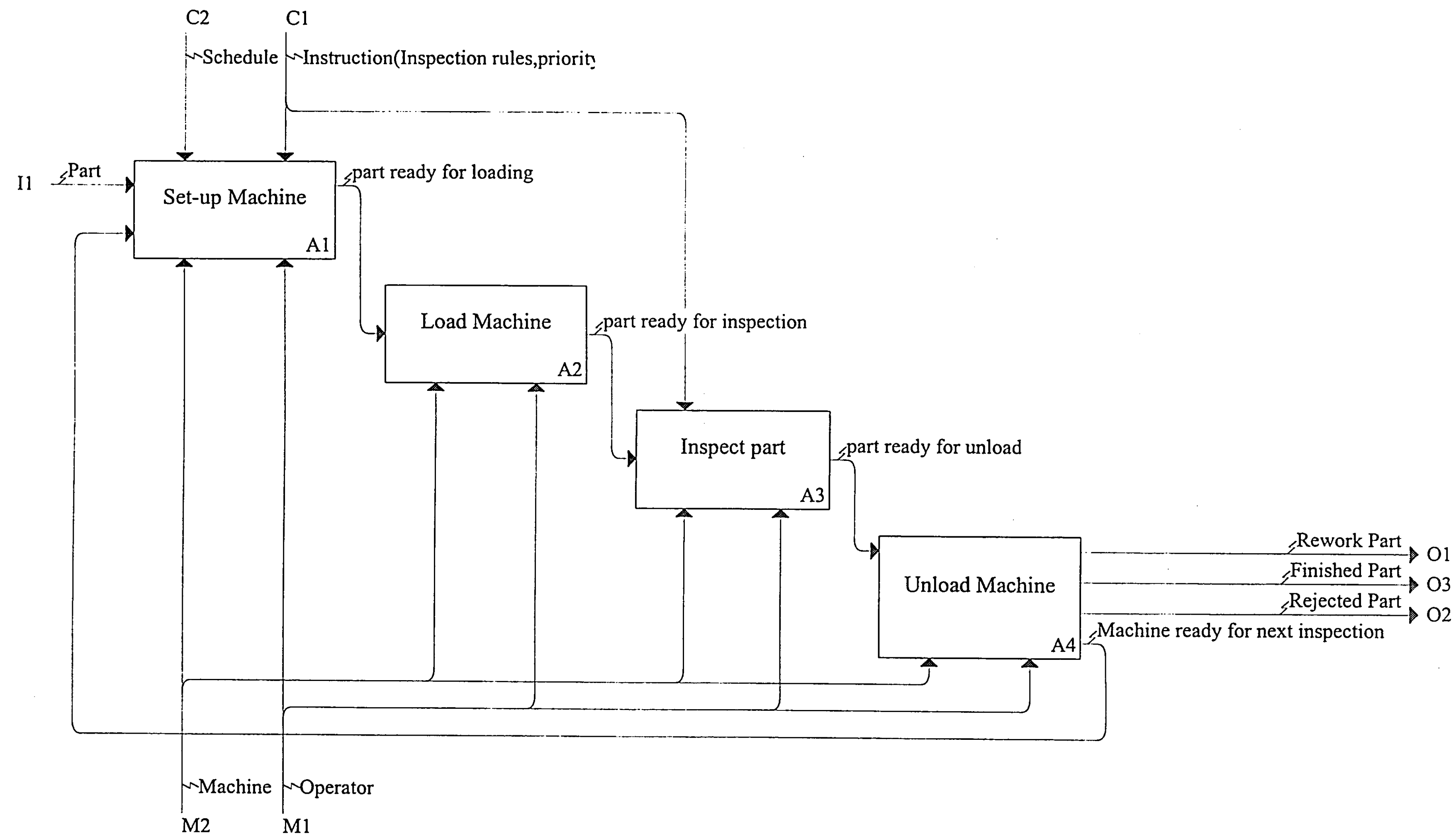
x Working
Draft
Recommended
Publication

READER | DATE | Context



Node: A-0 / C1 | Title: Module 2, Inspection (Context) | Number: Pg 1

Used At: Thesis	Author: Project: Activity Modules Library Notes: 1 2 3 4 5 6 7 8 9 10	Date: 8/28/98 Rev: 0 Time: 14:30:32	x Working Draft Recommended Publication	READER / DATE	Context

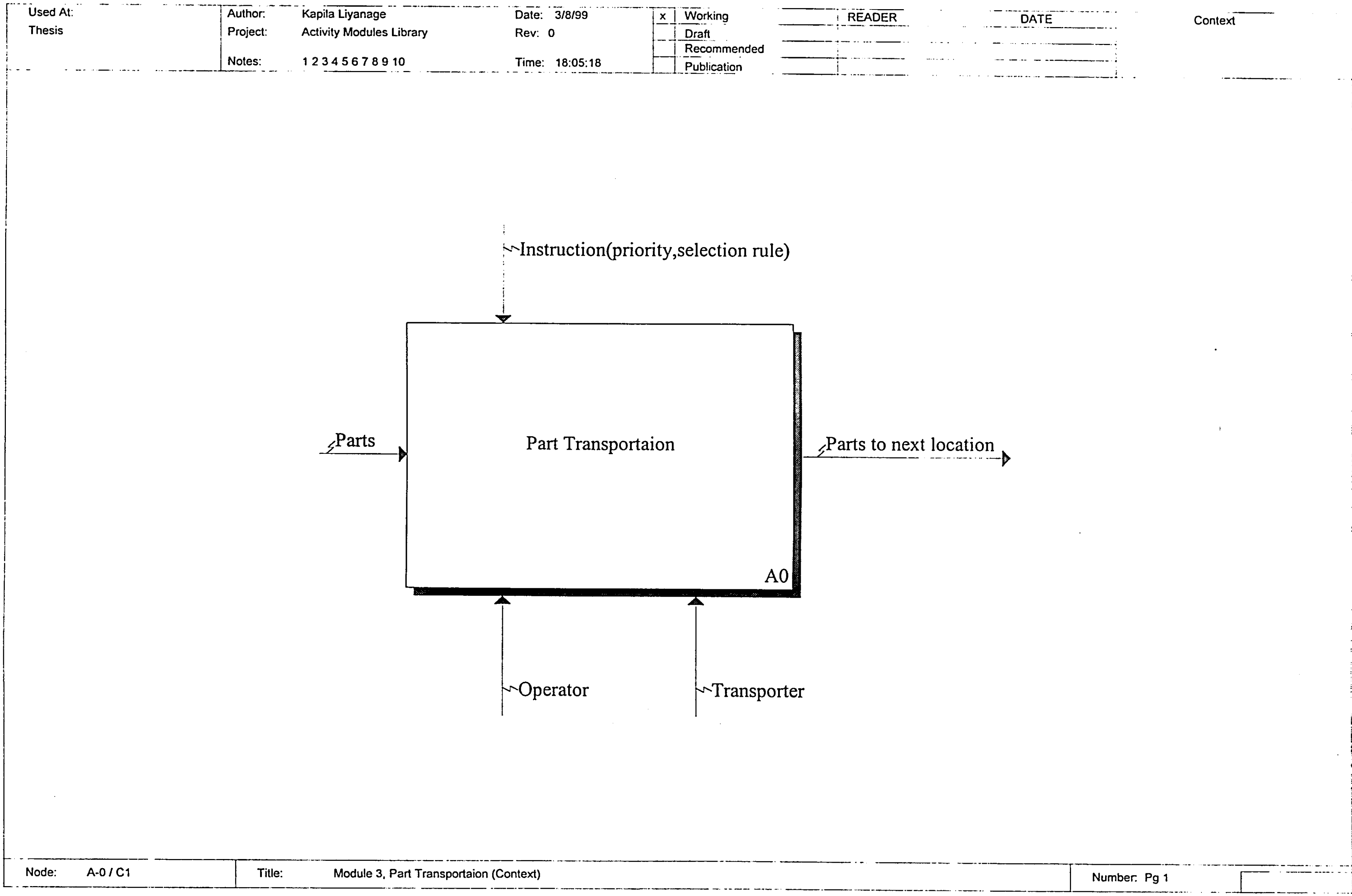


Node: A0 / C2	Title: Inspection	Number: Pg 2

Module 3

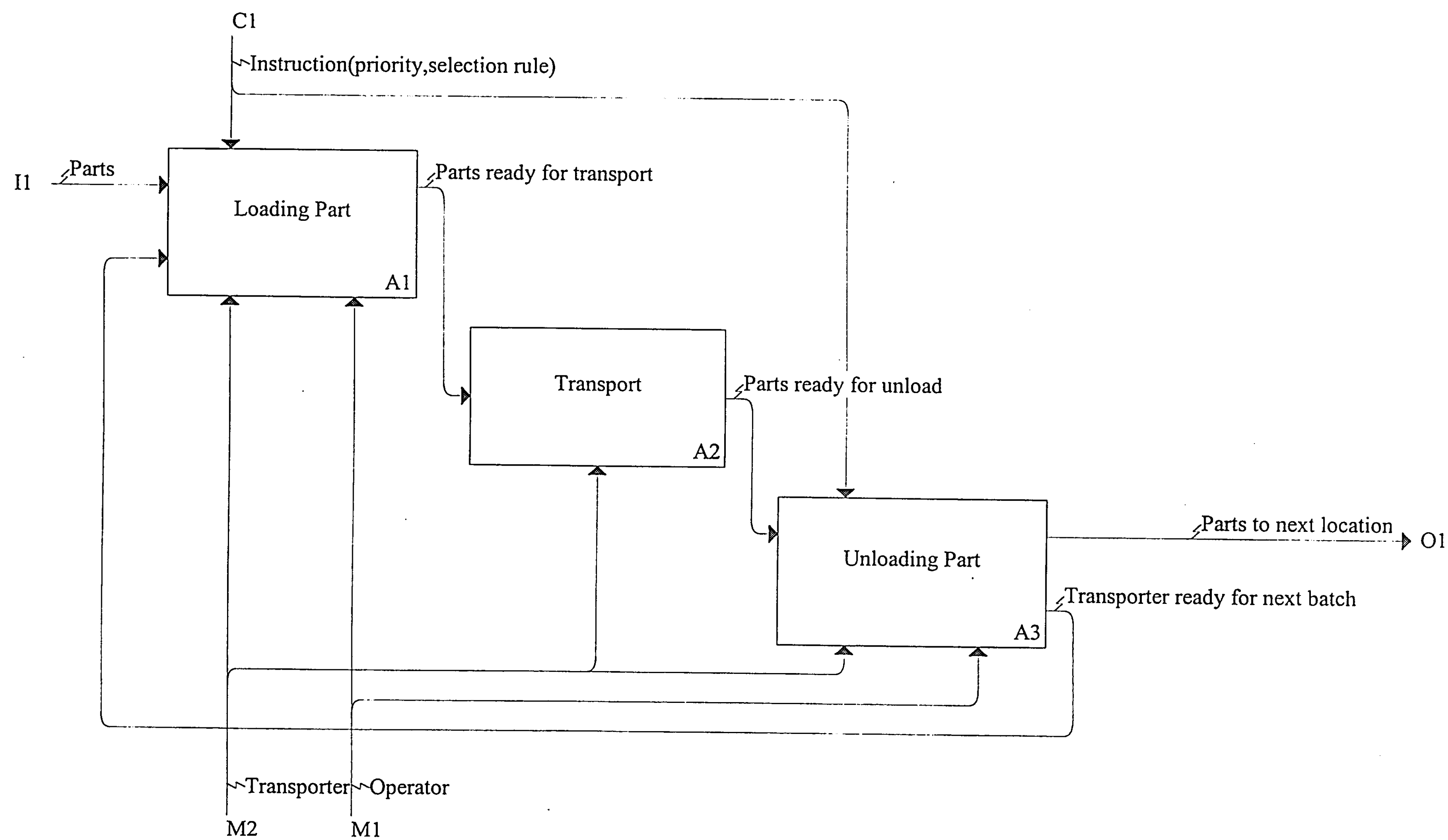
Material Handling

An important part of the manufacturing system is the movement of materials from one point to another (i.e. material handling systems). This diagram describes the required activities concerning material handling. The initial diagram (A0-Part transportation) is decomposed into three sub activities as shown in diagram.


Used At:
Thesis
Author: Kapila Liyanage
Project: Activity Modules Library
Notes: 1 2 3 4 5 6 7 8 9 10
Date: 3/8/99
Rev: 0
Time: 18:05:18
x Working
Draft
Recommended
Publication
READER
DATE
Context
Instruction(priority,selection rule)
Parts
Part Transportaion
Parts to next location
A0
Operator
Transporter
Node: A-0 / C1
Title: Module 3, Part Transportaion (Context)
Number: Pg 1

Used At: Thesis	Author:		
	Project: Activity Modules Library	Date: 1/7/99	x Working
		Rev: 0	Draft
	Notes: 1 2 3 4 5 6 7 8 9 10	Time: 12:36:08	Recommended
			Publication

READER DATE Context

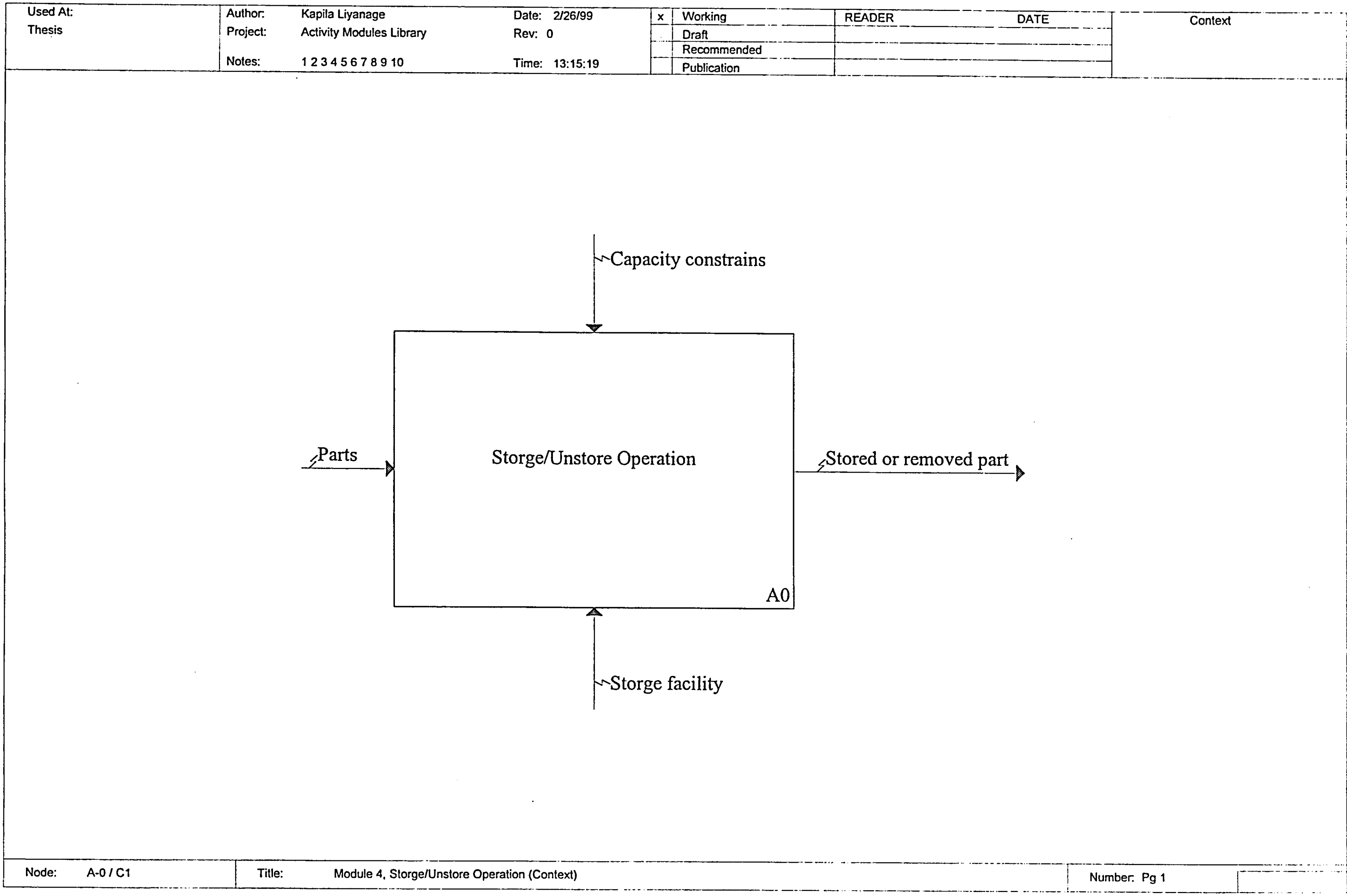


Node: A0 / C2 | Title: Part Transportaion | Number: Pg 2

Module 4

Storage operation

Storage is commonly used to hold (or store) parts. There are two types of storage operation. One operation is that parts add to the storage. The other one is parts remove from the storage (unstore). The basic assumptions in storage operation activity are; the parts are automatically added to the storage and automatically removed from the storage.


Used At:
Thesis
Author: Kapila Liyanage
Project: Activity Modules Library
Notes: 1 2 3 4 5 6 7 8 9 10
Date: 2/26/99
Rev: 0
Time: 13:15:19
x Working
Draft
Recommended
Publication
READER
DATE
Context
Capacity constrains
Parts
Storge/Unstore Operation
Stored or removed part
A0
Storge facility
Node: A-0 / C1
Title: Module 4, Storge/Unstore Operation (Context)
Number: Pg 1

4.4 PHASE 2: REFERENCE DATA MODEL (RDM)

4.4.1 Overview

In section 4.3 (Part 1), we examined and documented the generic activities of batch manufacturing systems, which are often used in simulation modelling. This section will review the reference data model for a MMOD approach. The reference data model defines the structure of data that is needed to support activities of a batch manufacturing system in a graphical form. The objective of a reference data model is to describe integration between various input data requirements for components, such as part, resources and system logic of a manufacturing system, into a single cohesive system, to describe a database implementation. This view of data is called the "conceptual schema".

Date (1990) explains that an architecture for a database management system is devised into three schemas known as the internal, external and conceptual schemas as illustrated in figure 4.6. An internal schema is defined in terms of file structures for storage and retrieval. An internal schema is often referred to as a storage view. A definition of the external schema is the representation of the data structure in a form appropriate (reports and screen design) to a user of the data. An external schema is often referred to as a user view. A conceptual schema is considered as an integrated data definition that is independent of any physical storage or external presentation format. The conceptual schema approach is required when the data is defined for the whole system, and not for a specific application inside it.

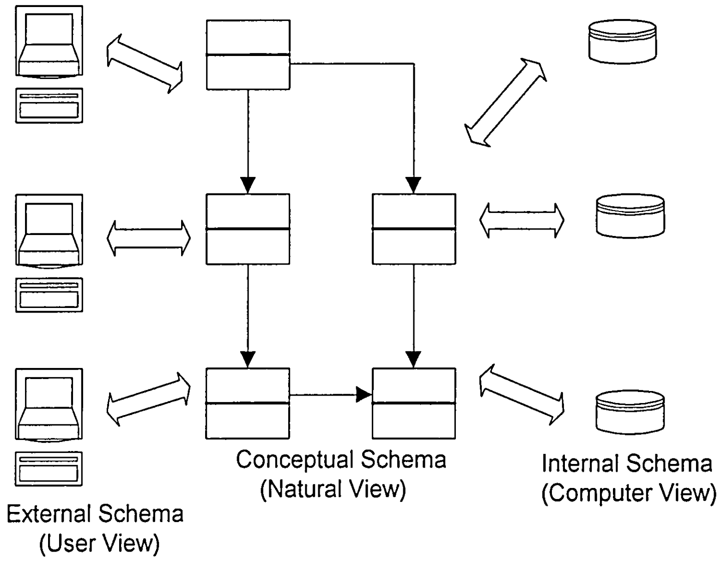


Figure 4.6. Three Schema Concept

In database design, the first step is the conceptual design of a database system. This is a fundamental description of the information that may be stored in the database. The primary objective of the conceptual schema is to provide a consistent definition of the meaning and interrelationship of data that can be used to integrate, share, and manage the integrity of data. A conceptual schema must have three important characteristics (Mayer 1992b):-

1. It must be consistence with the infrastructure of the business and be true across all application areas.
2. It must be extendible, such that , new data can be defined without altering previously defined data.
3. It must be transformable to both the required user views and to a variety of data storage and access structures.

4.4.2 IDEF1X Data Modelling

There are many methodologies available for data modelling, including the IDEF data modelling methodology as described in section 2.5. Having evaluated it in section 3.3, the IDEF1X data modelling methodology was selected to develop a reference data model. IDEF1X has been widely accepted by industry to model an information view of a manufacturing system. The main reasons for the use of IDEF1X in this research are the following characteristics;

- IDEF1X is a good graphical data modelling technique, used to represent the structure and semantics of data within the manufacturing system.
- IDEF1X model can represent a board range of details, therefore, they are suitable for supporting the full process of developing information systems.
- IDEF1X can support the development of the conceptual schema because its grammar can assure the semantic structure required by conceptual schema development, and a fully developed IDEF1X model may process the expected data consistency, extensibility and transformability.

Some other strengths of the IDEF1X methodology for data modelling were discussed in chapter 3.

There are many user friendly CASE tools available to support the IDEF1X data modelling approach based on its Entity-Relationship (E-R) diagrams. Most of the available CASE tools for IDEF1X data modelling are capable of automatic conversion from E-R diagrams to relational database schema through the use of their advanced software technology; This facility gives users greater flexibility in designing relational databases. For this research, SMARTER (KBSI, 1993), the intelligent information analysis and database design tool software, was used to develop the IDEF1X reference data model. SmartER automatically generates SQL code for database implementation and imports SQL for reverse engineering of databases into representative data models.

The reference data model shows the major entities (data groups) with their attributes and relationships. It currently contains 24 entities and over 75 attributes. This model can then be translated into a normalised relational database.

4.4.3 Model Syntax and Semantics

The reference data model depicts the data structure in a graphical form in terms of entities, their relationships and characteristics of entities (attributes). The various IDEF1X definitions are represented in this reference data model. This section presents a definition of the basic building blocks of an IDEF1X model (entity, attributes, and relationships). It then extends that discussion to different types of entities (independent and dependent), attributes (many key attributes and non-key attributes) and relationships (identifying, non-identifying and categorisation).By understanding syntax and semantics of IDEF1X, it assists the user to read a reference data model.

4.4.3.1 Entities

The entity is the most fundamental building block of IDEF1X data modelling. An entity is represented as a set of real things, such as a person, object, place, event, thing, etc. which have common attributes or characteristics. More precisely, an entity is a set or collection of like things called "instances"- (a single occurrence of an entity). For example, a reference model contains an entity such as a Part, Assembly-Structure, Machine, Material Handling Devices, Operation, etc. IDEF1X represents entities by rectangular boxes. An entity's name is recorded above its box. Entity names must be nouns or noun phrases. The entity may be assigned a reference tag number, which appears after the entity name and separated from it by a slash (Mayer, 1992b).

4.4.3.2 Identifier-Independent Entities & Identifier-Dependent Entities

In and IDEF1X model, entities are either identifier-independent (or simply "independent") or identifier-dependent (or simply "dependent") (Bruce 1992). Instances of identifier-independent entities can exist without any other entity instance, while instances of identifier-dependent entities are meaningless (by definition) without another associated entity instance. An IDEF1X represents independent entities by rectangular boxes and dependent entities by corner rounded boxes. For example, the rectangular box represents entity "customer" and the corner rounded box represents an entity "contact" (see figure 4.7).

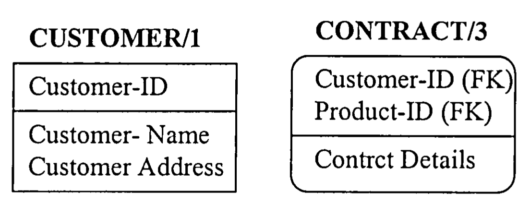


Figure 4.7 Symbols for dependent and independent entity

4.4.3.3 Attributes

The characteristic or properties of entity are called attributes. Attributes are shown inside the associated entity box. The primary key uniquely identifies an instance of an entity. For instance, each customer is uniquely identified by its primary key, consisting of the attribute of the "customer-ID". Attributes listed below the line are called non-key attributes. When a relationship exists between two entities (see definition of relationships), the primary key of the parent entity is inherited by the child as a foreign key. A foreign key is denoted by FK. For example, as shown in figure 4.8, the attribute, customer ID (primary key of the entity "customer") is inherited by the entity "contact" as a foreign key.

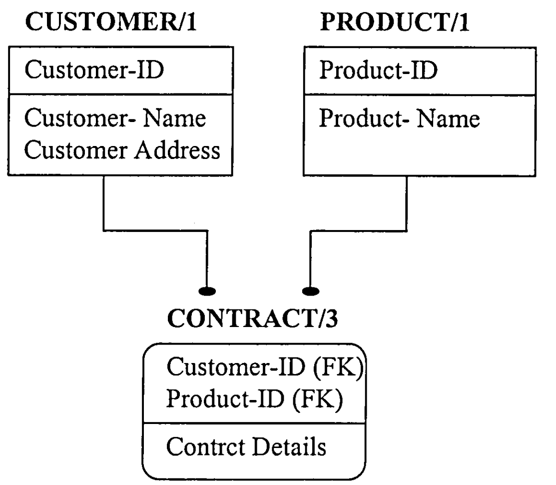


Figure 4.8 Entity representing an attribute Domain

4.4.3.4 Relationships

Relationships between entities are represented by lines between related entity boxes. Each line is liable with the relationship's name, which is a verb or verb phrase. The IDEF1X model represents two types of main relationships, namely, connection and generalisation relationships (Bruce, 1992 & Mayer, 1992b).

4.4.3.5 Connection Relationship

A connection relationship is an association between two entity instances. A connection is represented by a line terminated with a dot. This line is used to indicate the direction and cardinality. Each connection relationship has an associated cardinality. The cardinality specifies how many child entity instances may exist for each parent instance.

A connection relationship connects a line drawn between the parent entity and the child entity with a dot at the child end of the line. The default child cardinality is zero, one, or many. A "P" (for positive) is placed beside the dot to indicate a cardinality of one or more. A "Z" is placed beside the dot to indicate a cardinality of zero or one. If the cardinality is an exact number, a positive number is placed beside the dot (see figures 4.9 and 4.10).

4.4.3.6 Identifying and Non-identifying Relationships

If an instance of the child entity is identified by its association with the parent entity, the relationship is then referred to as an "identifying relationship". If every instance of the child entity can be uniquely identified without knowing the association of the parent entity, then the relationship is referred to as a "non-identifying relationship". Identifying relationships are indicated by a solid line between the parent and child entities, with a dot on the many (See figure 4.9). If an identifying relationship exits, the child entity is always an identifier-dependent entity, represented by a rounded corner boxes, and all the primary key attributes of a parent entity become (inherited) primary key attributes of a child entity.

A non-identifying relationship also connects a parent entity to a child entity. This is denoted by a dashed line (See figure 4.10). Both parent and child entities will be identifier-independent entities in a non-identifying relationship unless either or both are child entities in some other relationship which is an identifying relationship.

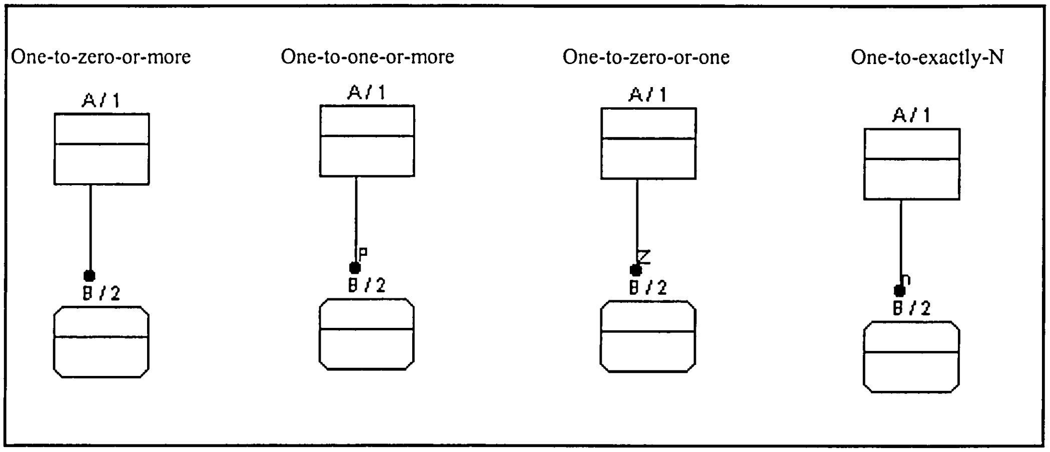


Figure 4.9 Identifying Relationships

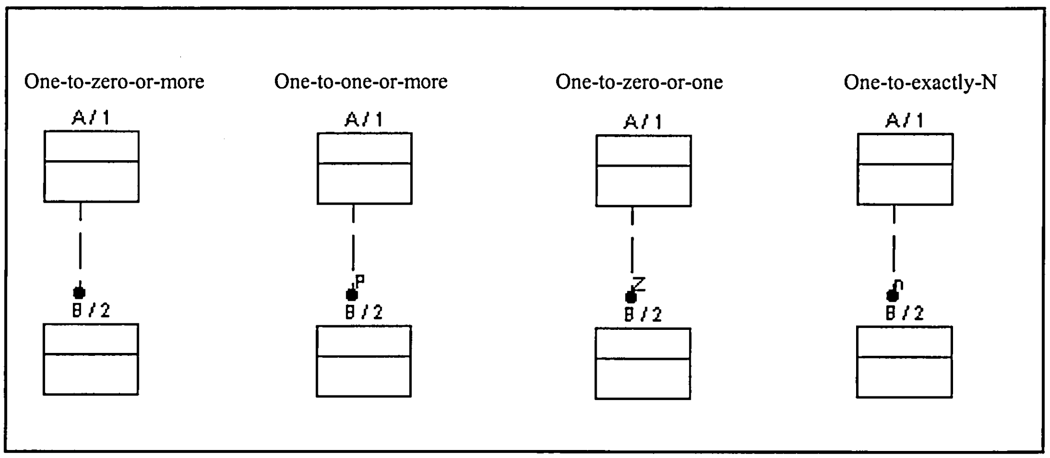


Figure 4.10 Non-identifying Relationships

4.4.3.7 Categorisation Relationships

Categorisation relationships allow the modeller to define the category of an entity. i.e. sub-type structure. Some real world things are categories of other real world things. As shown in figure 4.11, the employee (the generic parent) is either a Male-employee or a Female-employee (the categories).

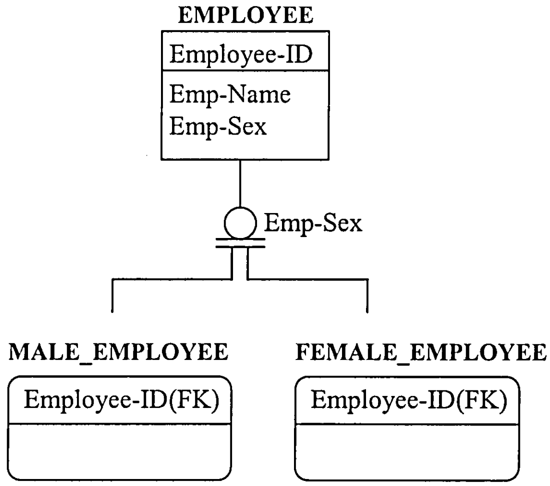


Figure 4.11 Categorisation Relationship

4.4.4 Structure of Reference Data Model (RDM)

Sections 4.4.3 introduced the most important rules and notation used in IDEF1X data modelling techniques and it will be very helpful to read and understand a developed reference data model (RDM).This section presents an overall construction of conceptual and physical schemas for a reference data model. To develop more reliable reference data model for a batch manufacturing environment, it very important to identify appropriate entities, their relationships, attributes and rules among the data structure, in terms of parts, resources and system logic, which are often used in simulation. Most of the entity and relationship definitions included in the reference data model are derived from the knowledge gained from activity modelling. The attributes needed for those entities were identified through literature, actual factory data and factory visits.

The data model is normalised into the third normal (Date, 1990). In this form, all the many-to-many relationships are resolved into one-to-many relationships by adding associative child entities between entities that have many-to-many relationships.

The reference data model consists of twenty four entities and more than seventy five attributes. Due to the complexity of the model, we divided it into nine subsets for a clear view. Each of these views link together and the overall reference model provides a data structure in a single view. The entire reference data model is divided into nine sub-set views as follows. Complete views for all entities and attributes can be found in data model (Refer Appendix A).

1. Part-Machine Operation (Non- assembly)
2. Bill of Material Structure (BOM)
3. Assembly Operation
4. Material Handling Operation
5. Inspection
6. Palletisation
7. Scheduling
8. Work cell-Machine-Machine Group
9. Storage Operation

View 1: Machine-Operation (Non-assembly operation)

Entity relationships between part, machine and operations are described on the reference model as follows. The machine can perform many operations and operations can be performed on many machines. An operation relates to many parts and parts can undergo many operations. In the Reference model, these many-to-many relationships can be resolved in three one-to-many relationships with associative entity **MACHINE_OPERATION**. The figure shows that a **MACHINE_ OPERATION** represents a three-way association among **PART** (Identified by Part-ID), **MACHINE** (Identified by Machine-ID), and **MACHINE _OPERATION** (Identified by Operation-No).

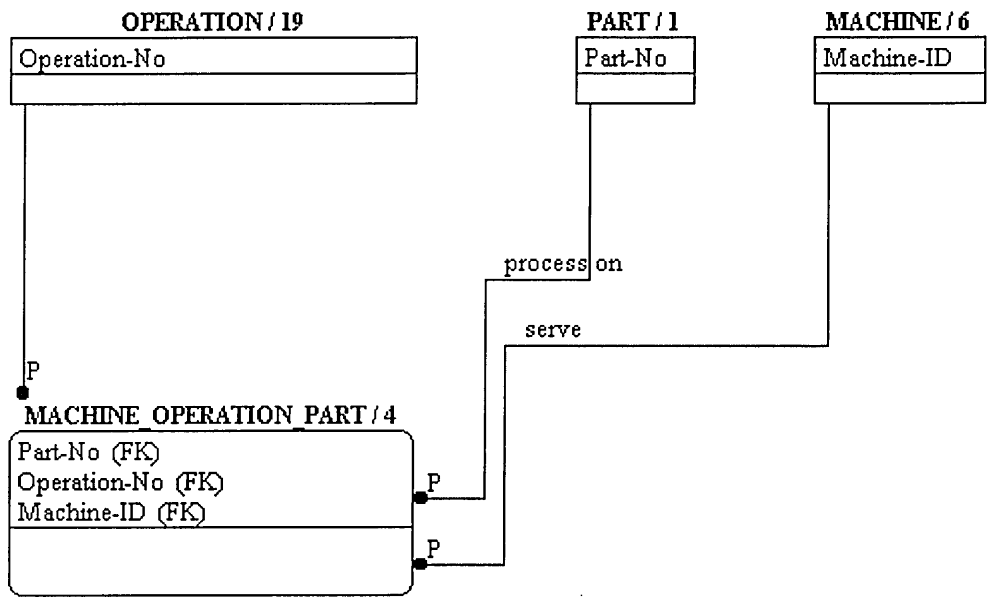


Figure 4.12 (View 1) Machine-Operation for non-assembly system

<u>**View 2: Bill of Material (BOM) Structure**</u>

A bill of material (BOM) structure can be represented by two entities, **PART** and **ASSEMBLY-STRUCTURE,** on the reference model. The entity **PART** has a dual relationship as a parent entity to the entity **ASSEMBLY-STRUCTURE** (Identified by Parent-component.Part-ID and Child-component.Part-ID). The same part sometimes acts as a component from which assemblies are made, i.e., a part may be a component in one or more assemblies, and sometimes, acts as an assembly, into which components are assembled, i.e., a part may be an assembly of one or more component parts (Mayer, 1992b). If the primary key for the entity **PART** is *Part-No*, then *Part-No* would appear twice in the entity **ASSEMBLY-STRUCTURE** as an inherited attributes. When a single attribute is inherited more than once, IDEF1X handles this situation by defining "role name" to each occurrence (Bruce, 1992). The role names of Parent component and Child component could be assigned to distinguish between the two inherited *Part-No* attributes.

Here are the definitions:-

Part-ID : The unique identifier of **PART**

Parent-Component.Part-ID : The identifier ("*Part-ID*") of the **PART** is from parent component of the **ASSEMBLY-STRUCTURE.**

Child-Component.Part-ID : The identifier ("Part-ID") of the **PART** is from child component of the **ASSEMBLY-STRUCTURE.**

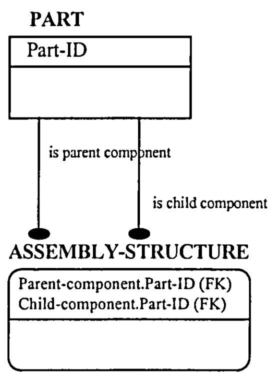
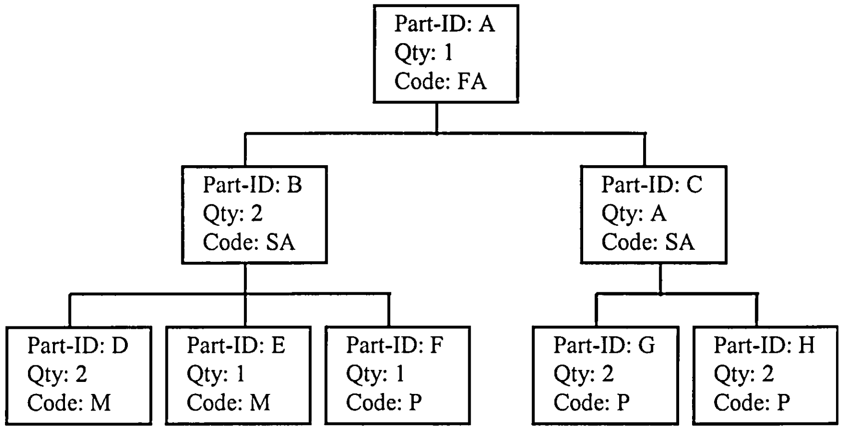


Figure 4.13 (View 2) Definition of bill of material (BOM) structure

For example, figure 4.14 illustrates the particular bill of material (BOM) structure and corresponding sample instance tables for entity **PART** and entity **ASSEMBLY_STRUTURE**, as described in tables 4.2 and 4.3 respectively.



FA : Final Assembly
SA : Sub Assembly
M : Manufactured
P : Purchased

Figure 4.14 BOM Structure

Part- ID	Part-Description	Part-Code	Required-Qty	Due-Date	Start-Date
A	Part A	FA			
B	Part B	SA			
C	Part C	SA			
D	Part D	M			
E	Part E	M			
F	Part F	P			
G	Part G	P			
H	Part H	P			

Table 4.2 Sample instance table for PART

Parent-Component.Part-ID	Child-Component.Part-ID	Quantity (For assembly)	Assy Level
A	B	2	1
A	C	1	1
B	D	2	2
B	E	1	2
B	F	1	2
C	G	2	2
C	H	2	2

Table 4.3 Sample instance tables for BOM structure

View 3: Assembly-Operation

This view describes a data model for assembly planning in a batch manufacturing system. The model used two special entities called, "**ASSEMBLY_TASK**" and "**ASSEMBLY_OPERATION**". as shown in figure 4.15.

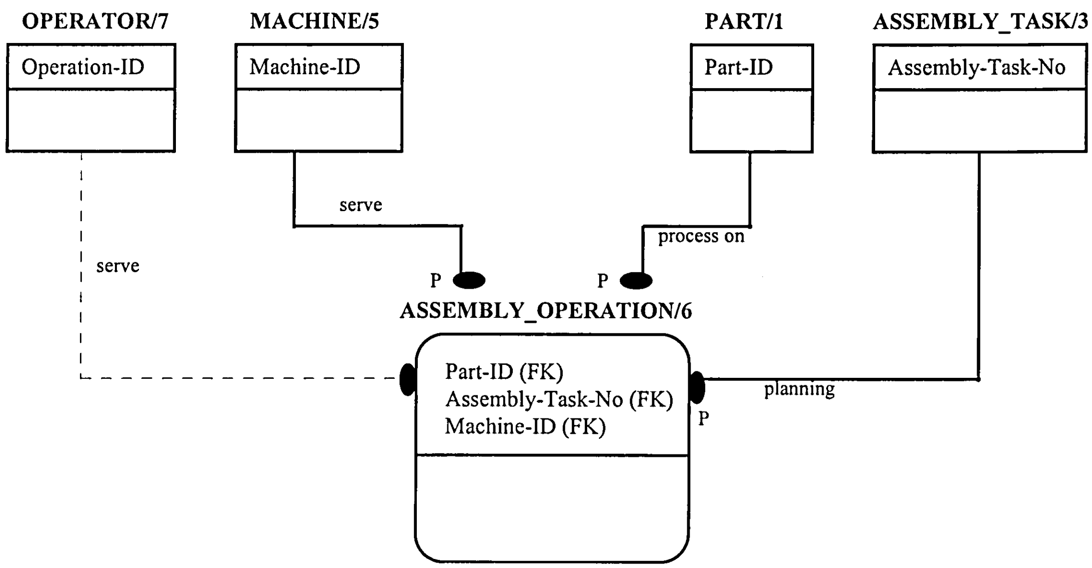


Figure 4.15 (View 3) Assembly-Operation

The entity "**ASSEMBLY_TASK**" describes the series of assembly tasks that join parts together in an assembly operation. For example, join part B and part C to make part A. For the above example, particular tasks may be

Task 1: Drill part B

Task 2: Cut part C

Task 3: Joint part B and C to Make part A

The entity "**ASSEMBLY_OPERATION**" describes the machine, required operations and parts used for each assembly task and also all other attributes (e.g. machine time, loading/unloading time, etc.) needed to perform each operations (see table 4.4).

Part-ID	Assembly-Task- No	Assembly-Task-Descrp	Machine-ID	
A	10	Drill PartB	DM1	
A	20	Screw Part C	SM1	
A	30	Joint Part B&C	AM1	

Table 4.4 Sample instance table for ASSEMBLY_OPERATION

View 4: Material Handling System

An important part of the manufacturing system is the movement of materials from one point to another (i.e. material handling systems). Our reference data model supports the identification and collection of relevant material handling data required for the simulation model. This figure provides a brief description of the material handling system, i.e. **PART**s can be transported by many **MH_DEVICE**s and **MH DEVICE**s can transport many **PART**s. In the Reference model, these many-to-many relationships can be resolved in two one-to-many relationships with an associative entity **TRANSFER_OPERATION** (identified by Beginning Stn, Ending Stn and Part No). The MH Device (the generic parent entity) is either a Conveyor, a Free-path transporter, or a Guided-path transporter (the categories).

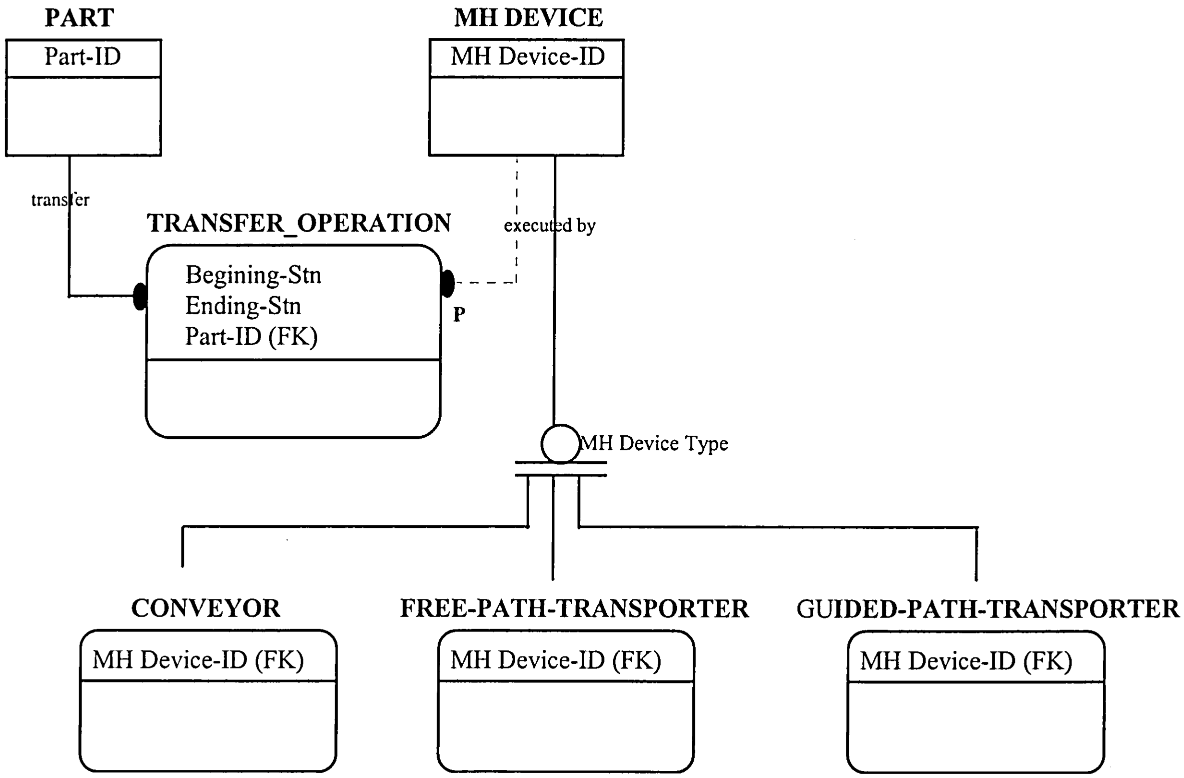


Figure 4.16 (View 4) Material handling operation

View 5: Inspection-Operation

The entity relationships for **INSPECTION_OPERATION** are similar to MACHINE_OPERATION relationships except for the special attributes defined in the **INSPECTION_OPERATION** entity. The two special attributes, "next operation for pass inspection" and "next operation for fail inspection", allows the collection of different destination data for passed and failed parts.

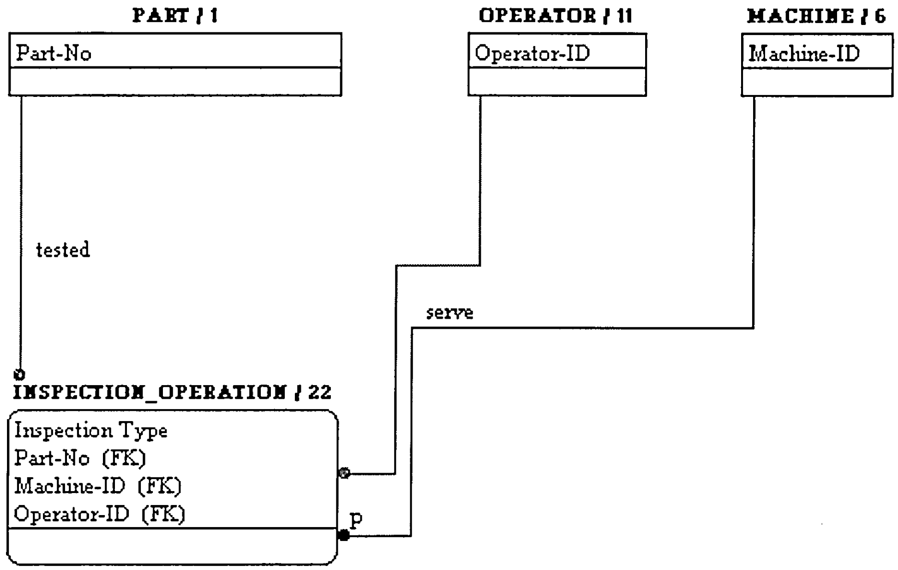


Figure 4.17 (View 5) Inspection Operation

View 6: Palletization

Parts are mounted on a pallet to be processed by a machine. A part can be mounted on different pallets and a pallet can be used for different parts. The figure 4.18 shows how the reference model describes this Palletisation operation.

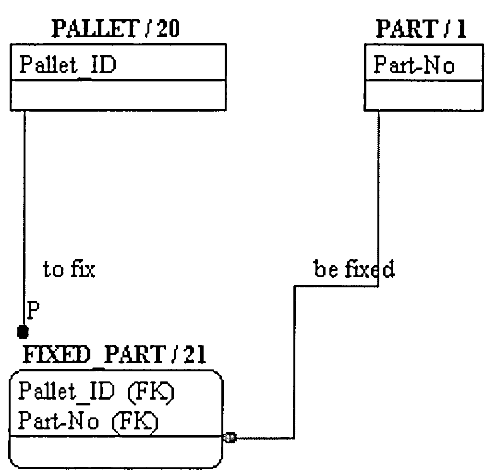


Figure 4.18 (View 6) Palletisation

View 7: Work Cell, Machine and Machine Group

This view (figure 4.19) consists of entity **WORK_CELL, MACHINE_GROUP** and **MACHINE**. The relationships indicates that a **WORK_CELL** (identified by Work-Cell-ID) contains many **MACHINEs,** but a particular **MACHINE** is located in only one **WORK_CELL.** Other relationships indicate that a **MACHINE** belongs to a certain **MACHINE_GROUP** (identified by Machine-Group-Name), and a **MACHINE_GROUP** may contain many **MACHINE**s.

View 8: Scheduling

Scheduling is the process of defining the available time duration (start and finish time information) for each resource. The entity-relationships diagram (figure 4.20) for Scheduling represents **MACHINE_SCHEDULE** and **OPERATOR_SCHEDULE** entities. Each entities defines the attributes for resource at available and unavailable times.

View 9: Storage View

This view (figure 4.21) explains the storage operations. The **PART**s are stored in the **STORAGE**s. The relationships indicate that a **PART** can be stored in different **STORAGE**s, and a **STORAGE** can be used for different **PART**s.

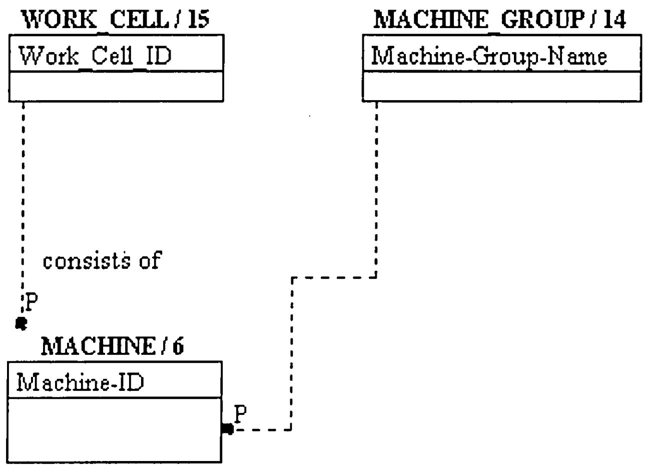


Figure 4.19 (View 7). Work cell, Machine Group and Machine allocation

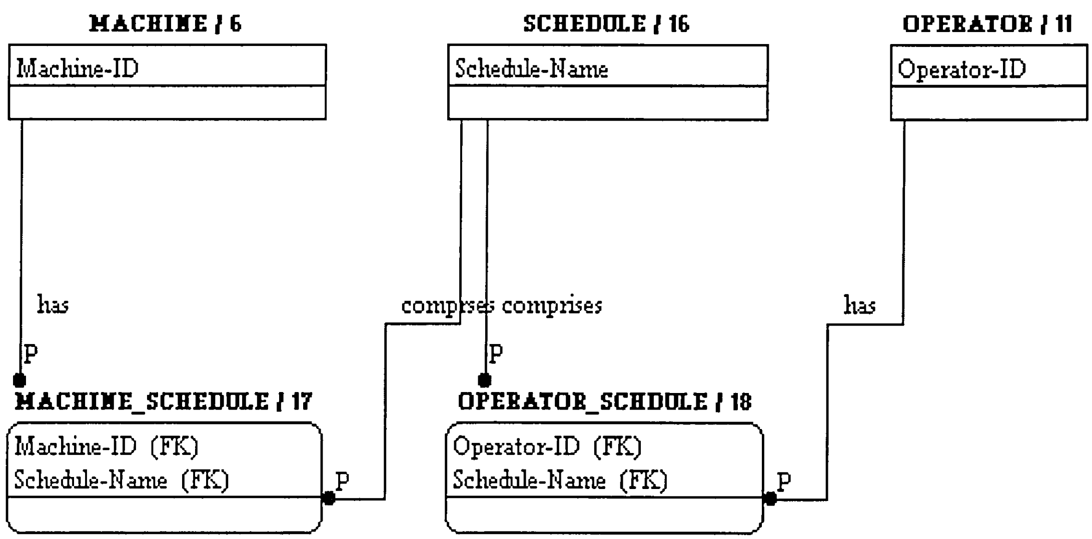


Figure 4.20 (View 8) Scheduling

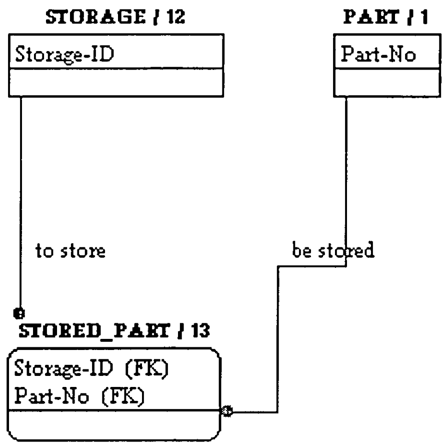


Figure 4.21 (View 9) Storage Operation

4.4.5 Transformation & SQL Code Generation

The resulting reference data model can be converted to a physical data model which shows exactly how the entities, relationships and attributes will be transformed into an actual database implementation. This section depicts the transformation of the detailed logical design of the reference data model into the physical design for the database.

Many of the CASE tools available for IDEF1X modelling can help in designing the data structure of a database and can automatically generate the SQL-DDL (Structured Query Language-Data Declaration Language) for the physical database model. The Data Declaration Language (DDL) of SQL contains commands for creating tables, indexes, and views (Date, 1994). These SQL DDL codes contain all of the necessary commands for creating tables, indexes and views (Boback, 1997). SmartER, which was used to develop a reference data model, can automatically generate SQL DDL codes for database implementation of commercial products such as Oracle, Microsoft SQL Server, dBASE and Microsoft Access, etc.

For a physical database, each entity in the reference data model (logical) becomes a table and each attribute becomes a column. Primary and foreign keys are declared for each table and referential integrity constraints are declared for each relationships.

As an example, here is the SQL DDL for a target database model for entity part, operation and machine-operation of the reference data model (see table 4.5).

```
CREATE TABLE PART
(
        Part_No CHAR(10)  NOT NULL,
        Part_Description  CHAR(10)  NOT NULL,
        Requird_Quantity          INTEGER  NOT NULL,
        Due_Date          DATE  NULL,
        Start_Date        DATE  NULL,
        PRIMARY KEY( Part_No )
);

CREATE UNIQUE INDEX XPKPART
        ON PART
(
        Part_No ASC
);
CREATE TABLE MACHINE
(
        Machine_ID        CHAR(10)  NOT NULL,
        Machine_Description       CHAR(10)  NOT NULL,
        Machine_Group_Name        CHAR(10)  NOT NULL,
        Work_Cell_ID      CHAR(10)  NULL,
        PRIMARY KEY( Machine_ID ),
        CONSTRAINT MACHINE_GROUP__MACHINE
        FOREIGN KEY( Machine_Group_Name)
        REFERENCES MACHINE_GROUP
        ON DELETE CASCADE,
        CONSTRAINT consists_of_
        FOREIGN KEY( Work_Cell_ID)
        REFERENCES WORK_CELL
        ON DELETE CASCADE
);

CREATE UNIQUE INDEX XPKMACHINE
        ON MACHINE
(
        Machine_ID        ASC
);
CREATE TABLE OPERATION
(
        Operation_No      CHAR(10)  NOT NULL,
        Operation_Description     CHAR(10)  NOT NULL,
        PRIMARY KEY( Operation_No )
);

CREATE UNIQUE INDEX XPKOPERATION
        ON OPERATION
(
```

```
        Operation_No     ASC
);

CREATE TABLE MACHINE_OPERATION_PART
(
        Machine_Time    NUMBER(6,2)  NOT NULL,
        Scrap_Rate      NUMBER(6,2)  NULL,
        Batch_Size      INTEGER  NOT NULL,
        Max_Batches     INTEGER  NULL,
        Machine_Loading_Time   NUMBER(6,2)  NULL,
        Machine_Unloading_Time         NUMBER(6,2)  NULL,
        Priority  CHAR(10)  NOT NULL,
        Operator_Responsibilites  CHAR(10)  NOT NULL,
        Set_up_Time     INTEGER  NULL,
        Set_up_Description        CHAR(10)  NOT NULL,
        Part_No CHAR(10)  NOT NULL,
        Operation_No    CHAR(10)  NOT NULL,
        Machine_ID      CHAR(10)  NOT NULL,
        Operator_ID     CHAR(10)  NOT NULL,
        PRIMARY KEY( Part_No,
                Operation_No,
                Machine_ID ),
        CONSTRAINT serve
        FOREIGN KEY( Machine_ID)
        REFERENCES MACHINE
        ON DELETE CASCADE,
        CONSTRAINT process_on
        FOREIGN KEY( Part_No)
        REFERENCES PART
        ON DELETE CASCADE,
        CONSTRAINT serve
        FOREIGN KEY( Operator_ID)
        REFERENCES OPERATOR
        ON DELETE CASCADE,
        CONSTRAINT OPERATION__MACHINE_OPERATION_PART
        FOREIGN KEY( Operation_No)
        REFERENCES OPERATION
        ON DELETE CASCADE
);

CREATE UNIQUE INDEX XPKMACHINE_OPERATION_PART
        ON MACHINE_OPERATION_PART
(
        Part_No ASC,
        Operation_No    ASC,
        Machine_ID      ASC
);
```

Table 4.5 Example of Code SQL DDL

4.4.6 Development of a Database

Database management systems have many advantages; it can almost meet the demands of simulation input data organisation and manipulation. A database can be used to store the data needed to define manufacturing activities and operations at any level of detail required by the simulation application. It can also be used to retrieve and manipulate the data using SQL-DML (Structured Query Language-Data Manipulation Language) according to the user requirements -when and where he or she requires.

Having reviewed an aspect of databases in terms of conceptual design and SQL DDL code for table design, the relational database management system (RDBMS) is implemented using the identified properties. The underlying structure of a database is a reference data model. For a generic batch manufacturing system, a relational database has been developed using an entire reference model, but in reality, all ranges of tables and fields contained in a database may not be necessary for a given simulation project. In a real situation, the required entity and range of attributes needed for a given entity are dependent on the type of objectives and defined level of details. Thus, steps for the development of customised databases for the system under investigation (through the simulation project life cycle) will be discussed in chapter 5.

For this research, the entity-relationship model (reference model) has been converted into a Microsoft Access Database. The MS Access also offers relationships between tables which are very similar to relationships provided in IDEF1X reference data model. The entity-relationship described in the reference data model implies the implementation of the database aspect. For a relational DBMS, each entity in the reference model becomes a relational table and each attribute becomes a column. Primary and foreign keys are declared for each table and referential integrity constraints are declared for each relationship.

4.5 PHASE 3: MAPPING TABLES

Due to a lack of cohesion between these two models, mapping tables are introduced to integrate both activity and data models within the common framework. The mapping tables are used to integrate both the activity model and reference data model within the common framework. This integration allows the modeller to identify system activities, corresponding information and data quickly. The mapping tables map the manufacturing activities defined in the library against the entity and attributes represented in the reference data model. This mapping process has also been extended to map corresponding ARENA (Systems Modelling Corporation, 1995) simulation software program constructs. This mapping table can be extended further to represent the modules of other simulation packages. The development of this mapping table into ARENA and other simulation packages will assist novice modellers to build their models.

The analysis of mapping tables are described as follows.

1. Mapping table for Machine Process module
    - ⇨ Non-assembly (Table 4.6)
    - ⇨ Assembly (Table 4.7)
2. Mapping table for Inspection module (Table 4.8)
3. Mapping table for material handling module (Table 4.9)
4. Mapping table for storage operation module (Table 4.10)

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part **(I)**	Part ID	PART	CREATE
		Part Description	PART	CREATE
		Batch Size	MACHINE_OPERATION	CREATE
		Max Batches	MACHINE_OPERATION	CREATE
		Inter Arrival Time	MACHINE_OPERATION	CREATE
	Machine **(M)**	Machine ID	MACHINE	RESOURCE/SEIZE
		Machine Description	MACHINE	RESOURCE/SEIZE
		MTBF	MACHINE_GROUP	RESOURCE
		MTTR	MACHINE_GROUP	RESOURCE
		Input Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
		Output Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
	Operator **(M)**	Operator ID	OPERATOR	RESOURCE/SEIZE
		Operator Description	OPERATOR	RESOURCE/SEIZE
		Efficiency	OPERATOR	
		Skills	OPERATOR	
		Learning Curve Effect	OPERATOR	
	Machine Operation (Non-assembly) **(F)**	Machine Operation Desc.	MACHINE_OPERATION	
		Machine Time	MACHINE_OPERATION	DELAY
		Operator Responsibilities	MACHINE_OPERATION	
	Schedule **(C)**	Schedule Name	SCHEDULE	RESOURCE-->Schedule block
		Schedule Description	SCHEDULE	RESOURCE-->Schedule block
		Schedule Time	SCHEDULE	RESOURCE-->Schedule block
		Duration	SCHEDULE	RESOURCE-->Schedule block
		Machine available time	MACHINE_SCHEDULE	RESOURCE-->Schedule block
		Machine unavailable time	MACHINE_SCHEDULE	RESOURCE-->Schedule block
		Operator available time	OPERATOR_SCHEDULE	RESOURCE-->Schedule block
		Operator unavailble time	OPERATOR_SCHEDULE	RESOURCE-->Schedule block
	Instruction **(C)**	Priority Rule	MACHINE_OPERATION	SEIZE

Table 4.6 Mapping table for a machine process module (Non-assembly) (key I-Input, M-Mechanism, C-Control and F-Function)

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
2	Described in Levels 1 +	Described in Levels 1 & 2		
	Set-up Machine **(F)**	Set-up Description	MACHINE_OPERATION	
		Set-up Time	MACHINE_OPERATION	DELAY/VARIABLE
	Load Machine **(F)**	Loading Time	MACHINE_OPERATION	DELAY
	Unload Machine **(F)**	Unloading Time	MACHINE_OPERATION	DELAY

Table 4.6 (continued) Mapping table for a machine process module (Non-assembly) (key I-Input, M-Mechanism, C-Control and F-Function)

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Parts **(I)**	Part ID	PART	CREATE
		Part Description	PART	CREATE
		Batch Size	MACHINE_OPERATION	CREATE
		Max Batches	MACHINE_OPERATION	CREATE
		Inter Arrival Time	MACHINE_OPERATION	CREATE
	BOM Structure **(C)**	Parent component ID	ASSEMBLY_STRUCTURE	CREATE/MATCH
		Child component ID	ASSEMBLY_STRUCTURE	CREATE/MATCH
		Quantity	ASSEMBLY_STRUCTURE	CREATE/MATCH
		Assembly-Level	ASSEMBLY_STRUCTURE	
	Assembly Planning **(C)**	Assembly Task No	ASSEMBLY_TASK	
		Assembly Task Descrp	ASSEMBLY_TASK	
	Machine **(M)**	Machine ID	MACHINE	RESOURCE/SEIZE
		Machine Description	MACHINE	RESOURCE/SEIZE
		MTBF	MACHINE_GROUP	RESOURCE
		MTTR	MACHINE_GROUP	RESOURCE
		Input Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
		Output Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
	Operator **(M)**	Operator ID	OPERATOR	RESOURCE/SEIZE
		Operator Description	OPERATOR	RESOURCE/SEIZE
		Efficiency	OPERATOR	
		Skills	OPERATOR	
		Learning Curve Effect	OPERATOR	
	Machine Operation (Assembly) **(F)**	Machine Operation Desc.	MACHINE_OPERATION	
		Machine Time	MACHINE_OPERATION	DELAY
		Operator Responsibilities	MACHINE_OPERATION	

Table 4.7 Mapping table for a machine process module (Assembly) (key I-Input, M-Mechanism, C-Control and F-Function)

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1 (continued)	Schedule **(C)**	Schedule Name	SCHEDULE	RESOURCE-->Schedule block
		Schedule Description	SCHEDULE	RESOURCE-->Schedule block
		Schedule Time	SCHEDULE	RESOURCE-->Schedule block
		Duration	SCHEDULE	RESOURCE-->Schedule block
		Machine available time	MACHINE_SCHEDULE	RESOURCE-->Schedule block
		Machine unavailable time	MACHINE_SCHEDULE	RESOURCE-->Schedule block
		Operator avialble time	OPERATOR_SCHEDULE	RESOURCE-->Schedule block
		Operator unavailble time	OPERATOR_SCHEDULE	RESOURCE-->Schedule block
	Instruction **(C)**	Priority Rule	MACHINE_OPERATION	SEIZE
2	Described in Levels 1 +	Described in Levels 1 & 2		
	Set-up Machine **(F)**	Set-up Description	MACHINE_OPERATION	
		Set-up Time	MACHINE_OPERATION	DELAY/VARIABLE
	Load Machine **(F)**	Loading Time	MACHINE_OPERATION	DELAY
	Unload Machine **(F)**	Unloading Time	MACHINE_OPERATION	DELAY

Table 4.7 (Continued) Mapping table for a machine process module(Assembly) (key I-Input, M-Mechanism, C-Control and F-Function)

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part (I)	Part ID	PART	CREATE
		Part Description	PART	CREATE
		Inter Arrival Time	INSPECTION_OPERATION	CREATE
	Machine (M)	Machine ID	MACHINE	RESOURCE/SEIZE
		Machine Description	MACHINE	RESOURCE/SEIZE
		MTBF	MACHINE_GROUP	RESOURCE
		MTTR	MACHINE_GROUP	RESOURCE
		Input Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
		Output Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
	Operator (M)	Operator ID	OPERATOR	RESOURCE/SEIZE
		Operator Description	OPERATOR	RESOURCE/SEIZE
		Efficiency	OPERATOR	
		Skills	OPERATOR	
		Learning Curve Effect	OPERATOR	
	Inspection (F)	Inspection Time	MACHINE_OPERATION	INSPECT or RESOURCE with CHANCE
		Failure Probability	INSPECTION_OPERATION	INSPECT or RESOURCE with CHANCE
		Next Operation for Pass Inspection	INSPECTION_OPERATION	INSPECT or RESOURCE with CHANCE
		Next Operation for Fail Inspection	INSPECTION_OPERATION	INSPECT or RESOURCE with CHANCE
		Operator Responsibilities	INSPECTION_OPERATION	
	Schedule (C)	Schedule Name	SCHEDULE	INSPECT/RESOURCE-->Schedule block
		Schedule Description	SCHEDULE	INSPECT/RESOURCE-->Schedule block
		Schedule Time	SCHEDULE	INSPECT/RESOURCE-->Schedule block
		Duration	SCHEDULE	INSPECT/RESOURCE-->Schedule block
		Machine available time	MACHINE_SCHEDULE	INSPECT/RESOURCE-->Schedule block
		Machine unavailble time	MACHINE_SCHEDULE	INSPECT/RESOURCE-->Schedule block
		Operator avialble time	OPERATOR_SCHEDULE	RESOURCE-->Schedule block
		Operator unavailable time	OPERATOR_SCHEDULE	RESOURCE-->Schedule block
	Instruction (C)	Priority Rule	INSPECTION	SEIZE

Table 4.8 Mapping table for a inspection module (key I-Input, M-Mechanism, C-Control and F-Function)

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
2	Described in Levels 1 +	Described in Level 1 +		
	Set-up Machine **(F)**	Set-up Description	INSPECTION_OPERATION	
		Set-up Time	INSPECTION_OPERATION	DELAY/VARIABLE
	Load Machine **(F)**	Loading Time	INSPECTION_OPERATION	DELAY
	Unload Machine **(F)**	Unloading Time	INSPECTION_OPERATION	DELAY

Table 4.8 (continued) Mapping table for a inspection module (key I-Input, M-Mechanism, C-Control and F-Function)

Level of Decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part (I)	Part ID	PART	CREATE
		Part Description	PART	CREATE
	Part Transportation (F)	Beginning Station	TRANSFER_OPERATION	TRANSPORT or SEGMENT or ROUTE
		Number of parts to Load at this Stn	TRANSFER_OPERATION	TRANSPORT
		Ending Station	TRANSFER_OPERATION	TRANSPORT
		Number of parts to Load at this Stn	TRANSFER_OPERATION	TRANSPORT or SEGMENT or ROUTE
		Distance	TRANSFER_OPERATION	DISTANCE
		Travel time	TRANSFER_OPERATION	ROUTE
		Operator Responsibilities	TRANSFER_OPERATION	
	Transporter (M)	Device ID	MH_DEVICE	TRANSPORTER
		Velocity	MH_DEVICE	TRANSPORTER
		MTBF	MH_DEVICE	TRANSPORTER
		MTTR	MH_DEVICE	TRANSPORTER
		Number of Transporter units	MH_DEVICE	TRANSPORTER
	Operator (M)	Operator ID	OPERATOR	RESOURCE
		Operator Description	OPERATOR	RESOURCE
		Efficiency		
		Skills		
		Learning Curve Effect		
2	Instruction (C)	Priority	TRANSFER_OPERATION	ALLOCATE
	Described in Levels 1 +	Described in Levels 1 +		
	Loading Part (F)	Loading Time	TRANSFER_OPERATION	DELAY
	Unloading Part (F)	Unloading Time	TRANSFER_OPERATION	DELAY

Table 4.9 Mapping table for a material handling module (key I-Input, M-Mechanism, C-Control and F-Function)

Level of decompositi on	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part (I)	Part ID	PART	CREATE
		Part Description	PART	CREATE
	Storage (M)	Storage ID	STORAGE	STORE/RESOURCE
		Storage Description	STORAGE	STORE/ RESOURCE
		Capacity	STORAGE	
	Store/Unstore Operation (F)	Nimbler of Parts to be Stored/Unstored	STORED_PART	STORE/ RESOURCE and BATCH
	Capacity Constrains (C)	Remaining Capacity	STORED_PART	

Table 4.10 Mapping table for a storage operation module (key I-Input, M-Mechanism, C-Control and F-Function)

4.6 Summary

Data modelling is a time consuming activity in simulation projects. The rapid development and deployment of simulation models are often constrained by issues surrounding input data modelling. The proposed methodology provides a tool kit to accelerate the data modelling exercise. It utilises a activity module library, a data reference model and a mapping table to create a database of required data. This methodology has been applied to generic batch manufacturing systems and it can be developed and applied to specific sectors more effectively. This MMOD methodology can be used to model existing (AS-IS), as well as proposed (To-BE) systems.

An activity model shows the activities of batch manufacturing systems which are often used in simulation modelling. The activity model provides a complete view of a system operation by showing an activity in relation to its input, output, control and mechanism. The developed activity modelling diagram library contains static activity models which do not capture any dynamics of the system. However, an activity modelling library is capable of understand a system and these static models are capable of providing the vast majority of the data needed to build the dynamic model. The diagram library describes the activities necessary to use a static representation of the system which can identify the primary input for simulation. However, most of the simulation models are dynamic. Static models attempt to provide a static representation of a dynamic system. Developed static activity modelling diagrams are capable of providing the vast majority of information or data needed to build a dynamic system simulation model.

A reference data model is used to understand the inherent nature of the data needed to support activities which are often used in simulation. The derived reference data model identifies data entities and the attributes to be collected for a simulation model. The model also captures the relationships and cardinality (business rules) in batch manufacturing systems. Then, the resulting logical reference data model can be converted into a physical database model, and more formally, the physical model specifies the application of the logical model.

As mentioned in section 3.3, one of the major shortcomings of IDEF methodology is the lack of cohesion between the activity and information model. Therefore, our developed approach of MMOD informally maintains relationships between the activity and information models to overcome the above problem. This informal relationship is completely manual. Activity and information models have been integrated using mapping tables through cross references between the models.

Chapter 5 explains an application of this methodology through a simulation life cycle. It mainly presents the steps for the development of a customised entity-relationship model according to the level of detail under system investigation.

CHAPTER 5

USE OF THE METHODOLOGY

5.1 Introduction

This chapter mainly explains the steps for the development of a customised entity model for the system under investigation. The development of a customised entity model requires comprehensive understanding of the various stages of a simulation project life cycle. Hence, the application of a MMOD concept to the simulation project through the simulation life cycle is described in this chapter. This chapter also describes other advantages of the proposed methodology.

5.2 Simulation life cycle

The life cycle of a simulation project is often presented as a series of steps. In an ideal simulation model building exercise, each step in the life cycle is completed before the next begins. However, recent literature indicates (ex. Tye 1997, Trybula 1994 and, Nordgren 1995) that many activities take place simultaneously and influence each other as described in Figure 5.1. At the initial stage, problem formulation, objectives definition, system investigation and data collection become mixed together.

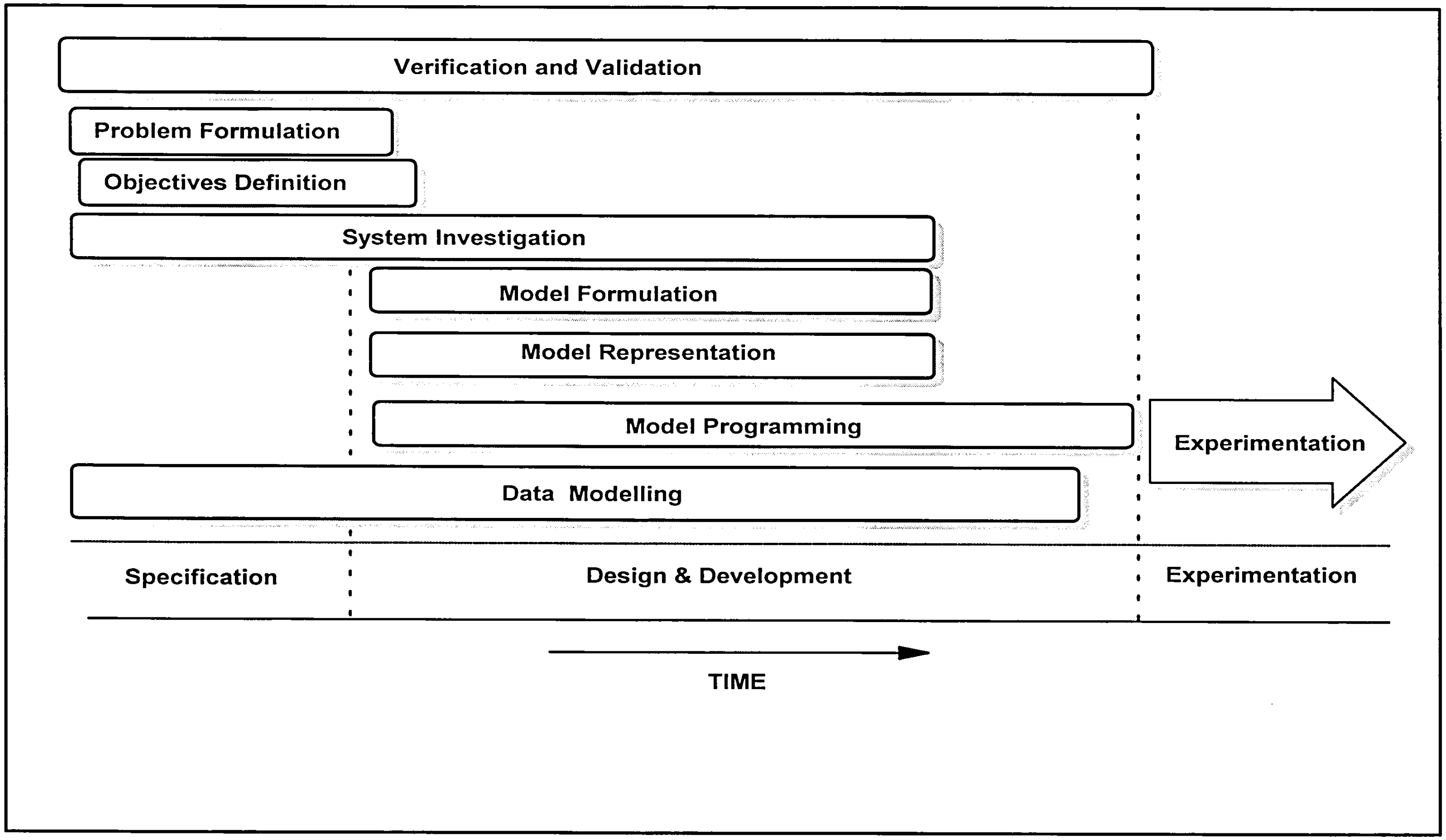


Figure 5.1 The simulation project activities (Based on :Tye & Perera, 1997)

5.3 Use of the methodology through the simulation life cycle

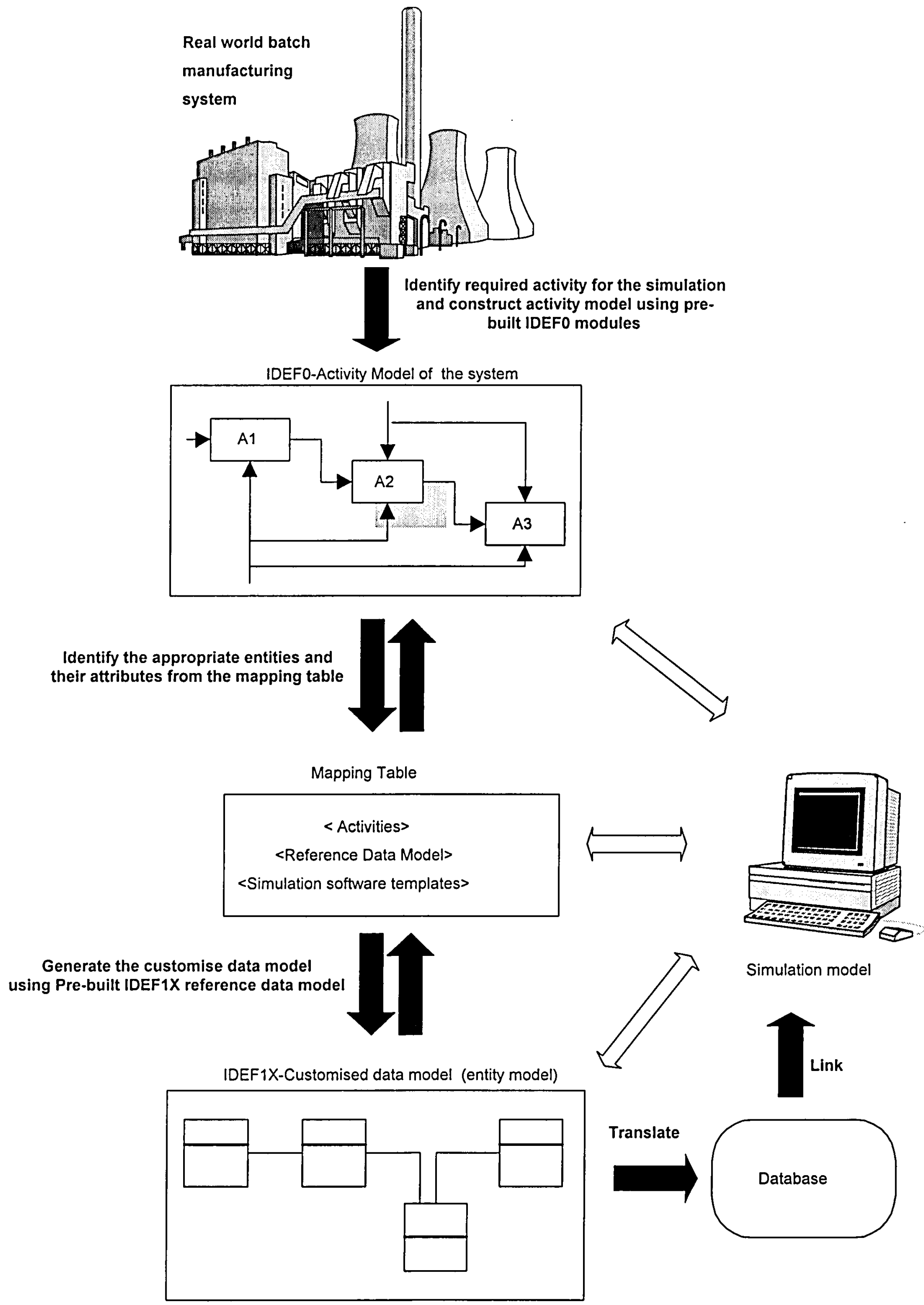


Figure 5.2 Use of the methodology

This section describes the application of a proposed methodology (Figure 5.2). The following steps should be carried out through a simulation life cycle to generate the required data model (entity model).

Step I: Formulate problem & Define Objectives.

One of the most important aspects of the simulation project is problem formulation. First, the problems needs to be identified at the initial kick-off meetings.

In order to identify problems, it is essential to understand the behaviour of the system to be simulated. The pre-built IDEF0 activity module library describes the structure and behaviour of system elements in terms of inputs, outputs, controls and mechanisms. Hence, the activity diagrams help in structuring ideas about the problems to be tackled and also communication between the participant in the simulation project. As described in section 3.2.1, it is always beneficial to facilitate communication and problem identification through graphical tools, because the right diagrams speak a thousand words.

Having formulated problems, the next step is to set objectives for a simulation project. Once the correct problem has been identified and the facts for simulation collected, the impact of the problem within its own boundaries is determined as well as the objectives to address the problems. As an example, when modelling a shopfloor system with the objective of determining machine utilisation, the machining time and set-up activities might be relevant within its own boundaries because they affect the utilisation of the machine.

Step II: Investigate the system.

At this stage, the modeller can review the system by completing a tour through the facility. Having reviewed the system, the modeller should have a rough idea of what is happening in the system. During this stage, all the system components such as parts, assembly parts and system resources, such as machine, operators, material handling devices with system logic, should be identified and recorded. A variety of methods, such as interviews with stakeholders, walk-through the system and use of operating manuals, are available to the model builder. At this stage, a pre-built IDEF0 activity modules library can be used to encourage the modeller to think of activities of the actual system. These diagrams illustrate the flow of parts through the system. Each decomposition level of activity diagrams help modellers to identify;

- What the inputs are required to perform each activity?
- What are the resources (machine, people, handling devices, etc.) used in each activity to perform the work ?
- What are the controls ?
- What are the outputs of the activities ?

Step III: Construct activity model of the system using pre-built IDEF0 modules.

Having listed all the system components, relevant activity diagrams are retrieved from the IDEF0 modules library. These activity modelling diagrams summarise the operational activities of system elements (such as part, resources, material handling devices, etc.) in manufacturing systems in a hierarchical manner. The main objective at this point is to develop a complete activity model of the system under investigation using pre-built IDEF logic modules. A library of modules relating to batch manufacturing systems is available to the model builder to create the reference model.

Generally, these modules are assembled in hierarchical fashion so that more details can be shown at lower levels. The result-in activity model forms the basis for identifying the data requirements. At the end of this step, a full conceptual diagram will be produced demonstrating how the system will operate and, which features should be included in the experiment model.

This result-in activity model will also aid the modeller by keeping it in mind for future reference, whenever he or she needs to review the facility again.

Step IV: Define Level of Model Details.

At this stage, the model builder can adjust the level of model details. Based on the objectives of the project, either further modules can be added to include more details or existing modules can be merged to decrease the level of details. As a rule of thumb, models should always include as little detail as possible in order to meet project objective(s). Time need not to be wasted initially collecting data that has no real impact on experimental factors; more data can always be gathered at a later stage.

Step V*:* Generate the required entity model using the mapping table.

The mapping table assists the modeller in identifying the appropriate entities and their attributes. The range of attributes required for a given entity is dependent on the level of decomposition defined in the activity model developed in Step III. The required entity model can be generated from the pre-built reference data model. It can be extendible or reducible, such that, new data can be added or previously defined data can be removed, without altering the previously defined data structure. The end result of this step is a customised entity model for the system under investigation. Most commercial packages available for IDEFIX modelling can automatically translate an entity model into a relational database.

Step VI: Collect data

The relational database, which consists of multiple data tables, defines the type of data to be collected. The order of data collection is governed by the rules of referential integrity. Further assistance will be provided to the model builder via a matching table which outlines the potential data sources for a given set of attributes. In an appropriate operating environment, it is possible to link data tables directly into data sources via the standard protocols such as ODBC.

For instance, the complete entity model has been converted into a Microsoft Access Database. The user-friendly data collection forms (screens) have been created for both input and output data. This includes a set of standard forms, which provide for the collection of required data for a simulation model and also retrieve collated data according to the user requirement using query language. The forms also serve as a dynamic link to the database. As an example, a few designed forms are described in figure 5.3.

Within the developed database system, it is always beneficial to design the data collection forms because these forms can be used to collect data, during actual interviews with stakeholders, walk through the systems and survey documentation.

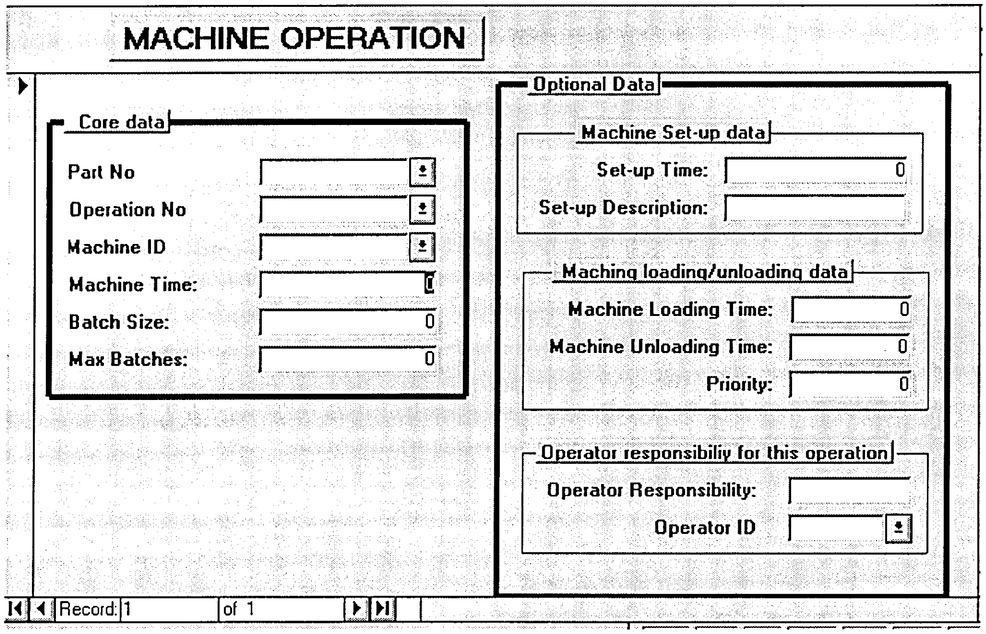


Figure 5.3 (a) Data collection forms to capture "Machine Operation" data

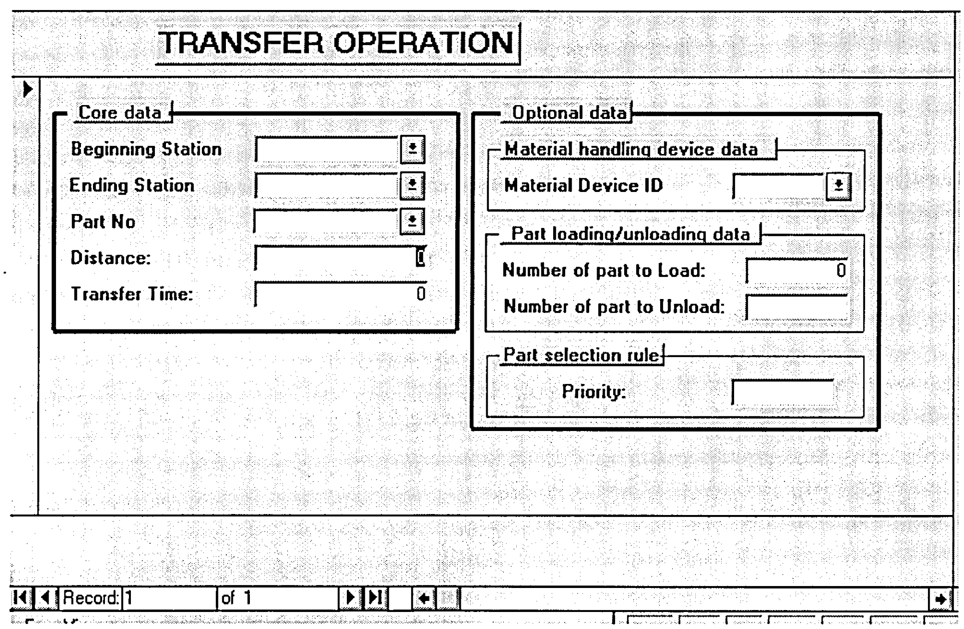


Figure 5.3 (b) Data collection forms to capture "Transfer" data

During data collection, the relevant forms can be retrieved from the database according to the type of answer the client presents. The form will then be completed in a sequential manner during the interview with the client or any other method. The reference database will guide the modeller through the correct steps to collect the required data during the interview. The order of correct steps for data collection is governed by the rules of referential integrity. As described in the reference data model, the database can enforce referential integrity, i.e. it protects users from adding or deleting data without violating the relationships. This reason simplifies things by giving a sequence of collecting data in a hierarchical manner as described in the following example in figure 5.4.

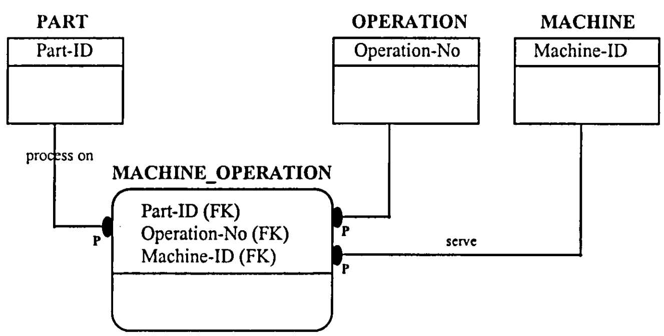


Figure 5.4. Example for sequence of collecting data

Sequence of inserting data:

Entity PART	No constraints to insert data
Entity OPERATION	No constraints to insert data
Entity MACHINE	No constraints to insert data
Entity MACHINE_OPERATION	No any MACHINE_OPERATION attributes may be inserted without PART, OPERATION, and MACHINE.

5.3.1 A Sample Application of methodology

This section presents an example that illustrates the use of the methodology. The objective of the example is to show how the required entity model is generated and how the created data is mapped into ARENA constructs.

This hypothetical example contains of a production cell which is a part of a manufacturing plant (Figure 5.5). This cell processes two types of parts. It consists of a CNC lathe, a grinding machine, a polishing machine and two operators. Operator 1 is responsible for loading, unloading of the lathe and the grinding machines. The lathe machine is required to set-up for different types of parts. They are then polished at the next station and operator 2 is required during the polishing process. The forklift delivers finished parts from the polish station to the assembly area.

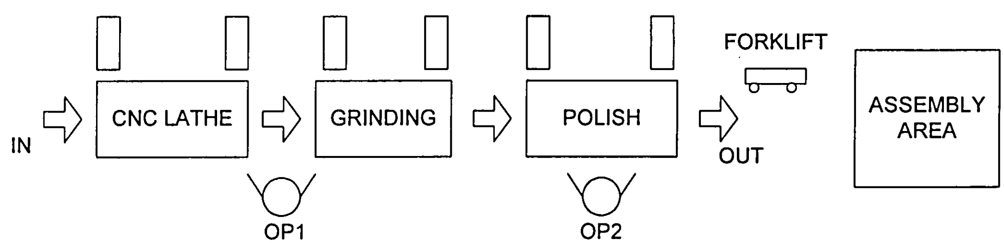


Figure 5.5 Production Cell

The parts enter the system one at a time with exponentially distributed time between arrivals, with a mean of 35 minutes for part 1, and 40 minutes for part 2. The process time for each parts are shown in table 5.1.

Part Type	CNC Lathe	Grinding	Polish
Part 1	Norm(13,1.5)	Norm(26,1.5)	Norm(15,0.5)
Part 2	Norm(26, 1.5)	Norm(28,1.5)	Norm(15,0.5)

Table 5.1 Process time

The loading/unloading times for both CNC and grinding machines are assumed to be normally distributed with a mean of 2 minutes and 0.5 standard deviation. The set-up time of CNC is normally distributed with 3.0 minutes and 0.5 standard deviation. The distance that forklift travels each way is about 20m. The average speed of the forklift is assumed to 10 meters/min.

In order to demonstrate the use of the proposed methodology, three scenarios with different level of details were developed using the above production cell.

Scenario 1. Level of Detail is Low

In this scenario, the processes at workstations are simply represented by a delay box. The parts enter the box, delay for a period of time and then leave the box. This is shown in figure 5.6.

Figure 5.6 Delay Box

Scenario 2. Level of Detail is Medium

In this scenario, the more details are added to increase the complexity of the model. At this level, the following factors are taken into consideration.

- Machine set-up

  In this scenario, CNC lathe is needed to set-up for different types of parts, thus requiring additional data such as set-up descriptions and set-up times.

- Machine loading/unloading

  To make the model more realistic, loading and unloading the machines are also modelled.

- Operator allocation

  Two operators (OP1 and OP2) are allocated to the system. OP1 is used for loading and unloading CNC lathe and the grinding machines. OP2 is exclusively used at polishing station.

- Transporter

  A transpoter is added to move finished components to assembly area. Modelling of a transporter requires further data such as velocity and distance.

- Queue priority

  The queue priority defines the selection rules associated with queues. It specifies the behavior of queues, such as First In-First Out (FIFO), Last In-First Out (LIFO) and Highest Value First (HVF).

Scenario 3. Level of Detail is High

Further factors are added into the scenario 3 to increase the complexity of the model. Factor such as machine breakdowns, buffers and scheduling details have been included in addition to the factors mentioned in scenario 2.

- Machine breakdown

  Breakdowns define failure characteristics such as time between down times and down time. In this example, the following breakdown data have been used (see table 5.2).

Resource	Time between failures (Min.)	Down time(Min.)
CNC Lathe	Exp(500)	Norm (10,5)
Grinding	Exp (1000)	Norm (10,2)
Polish	Exp (1500)	Norm (15,2)

Table 5.2 Breakdown details

- Input buffers

  Input buffers with limited capacity have been added for each machines in the production cell.

- Scheduling

  The scheduling defines the time-dependent capacity changes. In this example, all the resources are following 25 minutes break for every 2500 minutes.

Before the construction of the entity model, it is necessary to review the activity models as the range of attributes and entities required for the entity model depends on the components and decomposition levels of the activity models.

The following steps are detailed for each scenario with appropriate diagrams, tables and descriptions.

Step I. Development of activity models

Having identified required elements, the pre-built IDEF0 activity modules are assembled in a hierarchical manner to construct an activity model of the system. The constructed activity models are illustrated in following figures.

- Figure 5.7 Activity model for scenario 1
- Figure 5.11 Activity model for scenario 2
- Figure 5.15 Activity model for scenario 3

Step II. Selection of appropriate mapping tables and required data

For each element in the developed activity models, the required elements should then be identified from the mapping tables. By comparing identified elements from mapping tables, the required data items (attributes) and corresponding reference model entities can be adapted. The selected mapping tables highlight the required elements and data items as shown in following.

- Table 5.3 (a) Mapping table for machine process module (non-assembly)-scenario 1
  Table 5.3 (b) Mapping table for material handling module-scenario 1
- Table 5.5 (a) Mapping table for machine process module (non-assembly)- scenario 2
  Table 5.5 (b) Mapping table for material handling module-scenario 2

- Table 5.7 (a) Mapping table for machine process module (non-assembly)-scenario 3

  Table 5.7 (b) Mapping table for material handling module - scenario 3

Step III. Generation of the required entity model

Having identified appropriate entities and attributes from the steps II, a customised entity model is derived for the selected scenario using the generic reference data model as illustrated in following figures.

- Figure 5.8 The required entity model for scenario 1
- Figure 5.12 The required entity model for scenario 2
- Figure 5.16 The required entity model for scenario 3

The customised entity model forms the foundation for a computer database system. Each entity in the entity model becomes a table and each attribute becomes a column.

Step IV Construction of simulation model

Simulation models for each scenarios were constructed using Arena. Flow diagrams for logic are shown in following figures.

- Figure 5.9 Arena simulation model for scenario 1
- Figure 5.13 Arena simulation model for scenario 2
- Figure 5.17 Arena simulation model for scenario 3

Step V Mapping between the generated entity models and Arena templates

At this step, data from tables are mapped into appropriate Arena constructs. A sample of Arena constructs for each scenario are shown in following figures.

- Figure 5.10 Data mapping between entity model and constructed arena model for scenario 1
- Figure 5.14 Data mapping between entity model and constructed arena model for scenario 2
- Figure 5.18 Data mapping between entity model and constructed arena model for scenario 3

Step VI. Example outputs of the constructed simulation models

The models were run for 15000 minutes, with three replications and 3000 minutes warm-up period to see the key result such as throughput times, resources utilisation, individual queues and, downtimes , etc. The results for simulation model are shown in summary reports. Refer following tables.

- Table 5.4 Sample simulation outputs for scenario 1
- Table 5.6 Sample simulation outputs for scenario 2
- Table 5.8 Sample simulation outputs for scenario 3

It is obvious that the reference data model can be used to generate required entity models for each case. The generic reference data model defines the structure of data that is needed to support almost all batch manufacturing activities which are often used in simulation. The model also defines all the entity relationships and cardinality (business rules) in a manufacturing system. To generate the required entity models, the best way is to copy the generic reference data model. It can then be reduced to the required level by removing entities and attributes without altering the previously defined structure. All the entities and attributes are stored in the entity and attribute pools. More entities and attributes can be added to generate more complex entity models without changing the structure. The derived entity model can then automatically be translated into a relational database which can be used to collect and store required data for a simulation model.

Scenario 1

Level of detail is low

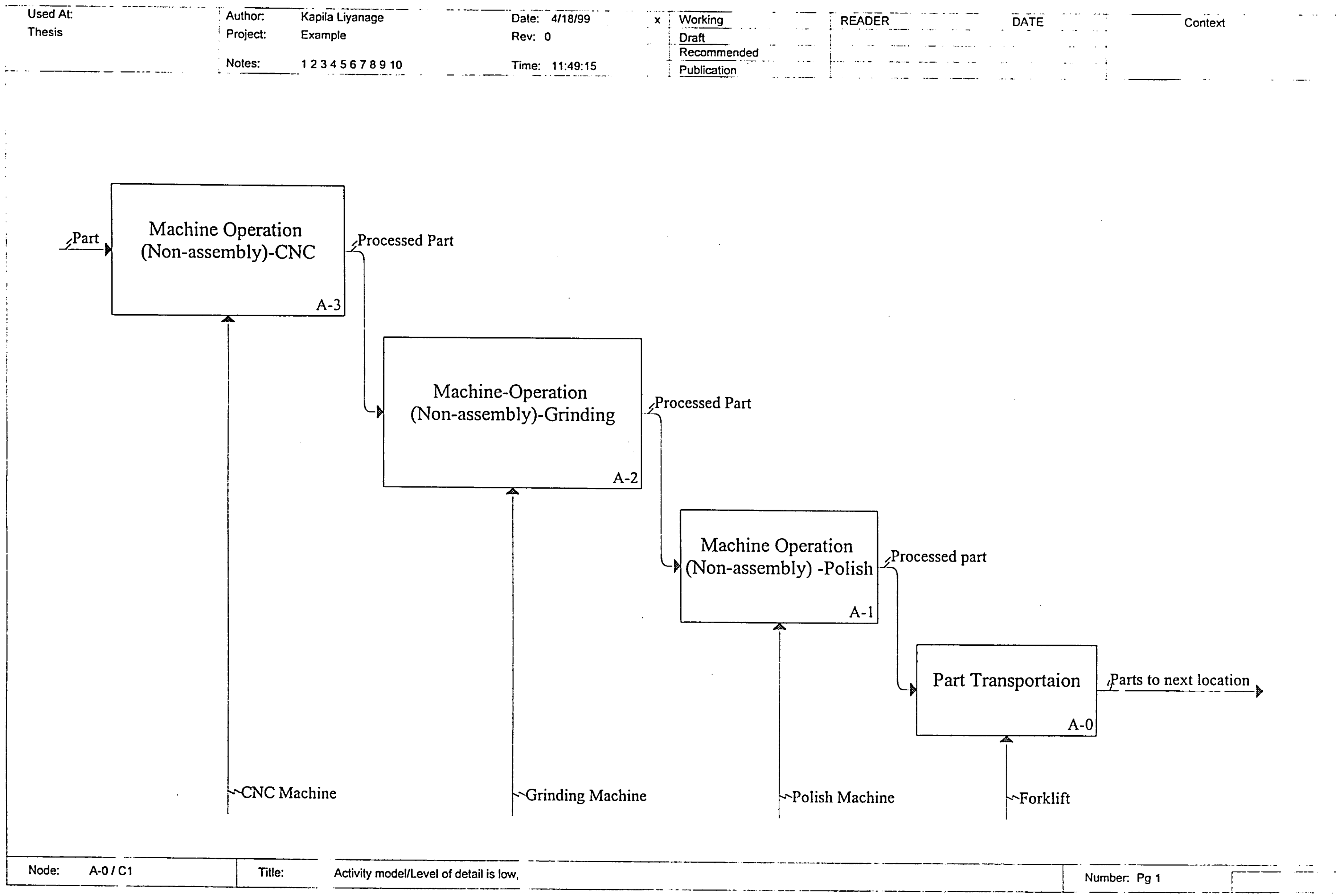


Figure 5.7 Activity model for scenario 1

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part **(I)**	Part ID	PART	CREATE
		Part Description	PART	CREATE
		Batch Size	MACHINE OPERATION	CREATE
		Max Batches	MACHINE OPERATION	CREATE
		Inter Arrival Time	MACHINE OPERATION	CREATE
	Machine **(M)**	Machine ID	MACHINE	RESOURCE/SEIZE
		Machine Description	MACHINE	RESOURCE/SEIZE
		MTBF	MACHINE GROUP	RESOURCE
		MTTR	MACHINE GROUP	RESOURCE
		Input Buffer Capacity	MACHINE GROUP	RESOURCE or STORAGE
		Output Buffer Capacity	MACHINE GROUP	RESOURCE or STORAGE
	Operator **(M)**	Operator ID	OPERATOR	RESOURCE/SEIZE
		Operator Description	OPERATOR	RESOURCE/SEIZE
		Efficiency	OPERATOR	
		Skills	OPERATOR	
		Learning Curve Effect	OPERATOR	
	Machine Operation (Non-assembly) **(F)**	Machine Operation Desc.	MACHINE OPERATION	
		Machine Time	MACHINE OPERATION	DELAY
		Operator Responsibilities	MACHINE OPERATION	
	Schedule **(C)**	Schedule Name	SCHEDULE	RESOURCE-->Schedule block
		Schedule Description	SCHEDULE	RESOURCE-->Schedule block
		Schedule Time	SCHEDULE	RESOURCE-->Schedule block
		Duration	SCHEDULE	RESOURCE-->Schedule block
		Machine available time	MACHINE SHEDULE	RESOURCE-->Schedule block
		Machine unavailable time	MACHINE SHEDULE	RESOURCE-->Schedule block
		Operator available time	OPERATOR SHEDULE	RESOURCE-->Schedule block
		Operator unavailable time	OPERATOR SHEDULE	RESOURCE-->Schedule block
	Instruction **(C)**	Priority Rule	MACHINE OPERATION	SEIZE

Table 5.3(a) Mapping table for scenario 1 [machine process module (Non-assembly)]

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
2	Described in Levels 1 +	Described in Levels 1 & 2		
	Set-up Machine **(F)**	Set-up Description	MACHINE_OPERATION	
		Set-up Time	MACHINE_OPERATION	DELAY/VARIABLE
	Load Machine **(F)**	Loading Time	MACHINE_OPERATION	DELAY
	Unload Machine **(F)**	Unloading Time	MACHINE_OPERATION	DELAY

Table 5.3(a) (continued) Mapping table for scenario 1 [machine process module (Non-assembly)

Level of Decomposition	Activity_Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part **(I)**	Part ID	PART	CREATE
		Part Description	PART	CREATE
	Part Transportation **(F)**	Beginning Station	TRANSFER_OPERATION	TRANSPORT or SEGMENT or ROUTE
		Number of parts to Load at this Stn	TRANSFER_OPERATION	TRANSPORT
		Ending Station	TRANSFER_OPERATION	TRANSPORT
		Number of parts to Load at this Stn	TRANSFER_OPERATION	TRANSPORT or SEGMENT or ROUTE
		Distance	TRANSFER_OPERATION	DISTANCE
		Travel time	TRANSFER_OPERATION	ROUTE
		Operator Responsibilities	TRANSFER_OPERATION	
	Transporter **(M)**	Device ID	MH_DEVICE	TRANSPORTER
		Velocity	MH_DEVICE	TRANSPORTER
		MTBF	MH_DEVICE	TRANSPORTER
		MTTR	MH_DEVICE	TRANSPORTER
		Number of Transporter units	MH_DEVICE	TRANSPORTER
	Operator **(M)**	Operator ID	OPERATOR	RESOURCE
		Operator Description	OPERATOR	RESOURCE
		Efficiency		
		Skills		
		Learning Curve Effect		
2	Instruction **(C)**	Priority	TRANSFER_OPERATION	ALLOCATE
	Described in Levels 1 +	Described in Levels 1 +		
	Loading Part **(F)**	Loading Time	TRANSFER_OPERATION	DELAY
	Unloading Part **(F)**	Unloading Time	TRANSFER_OPERATION	DELAY

Table 5.3 (b) Mapping table for scenario 1 [material handling module]

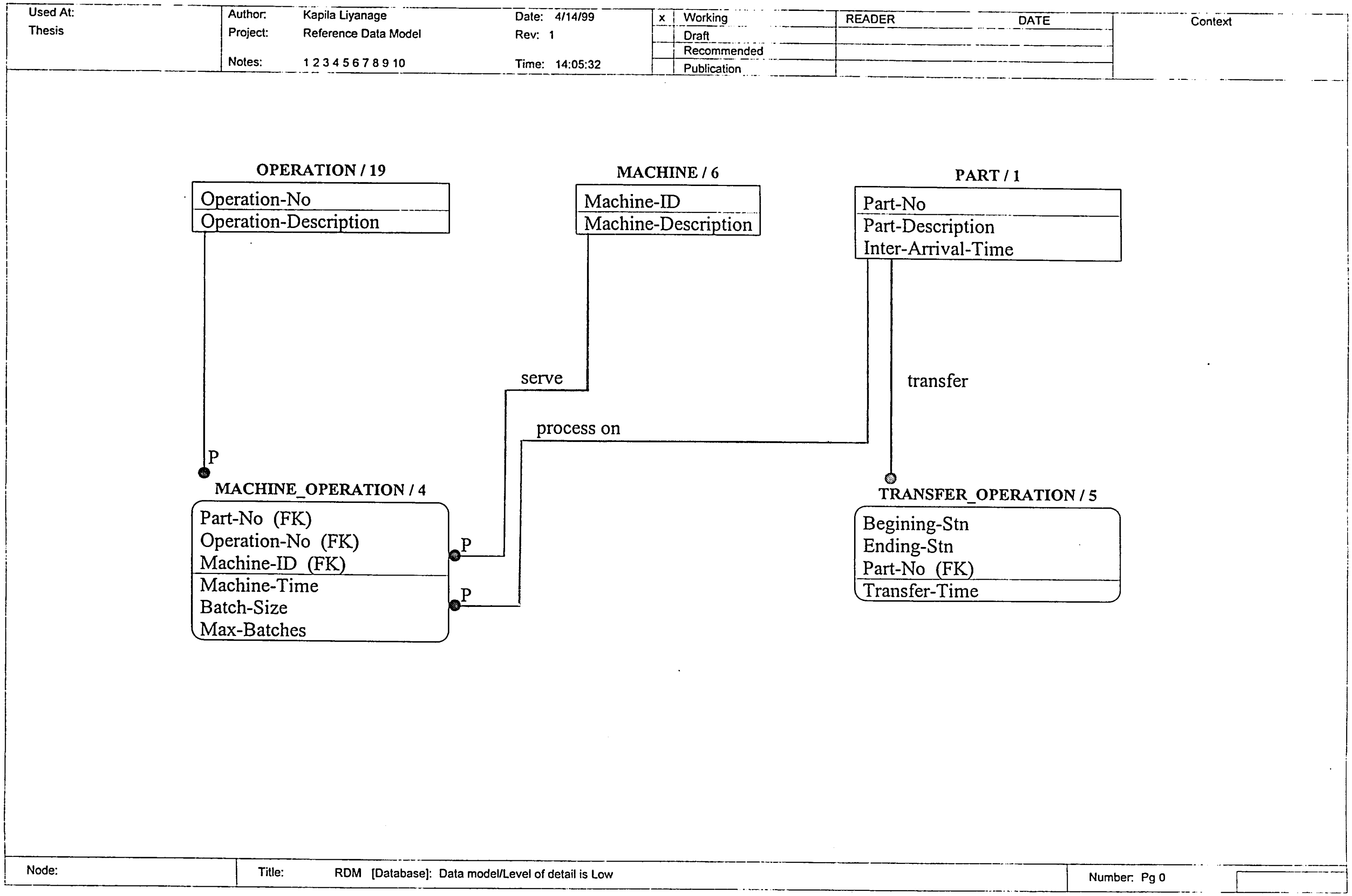


Figure 5.8 The required entity model for scenario 1

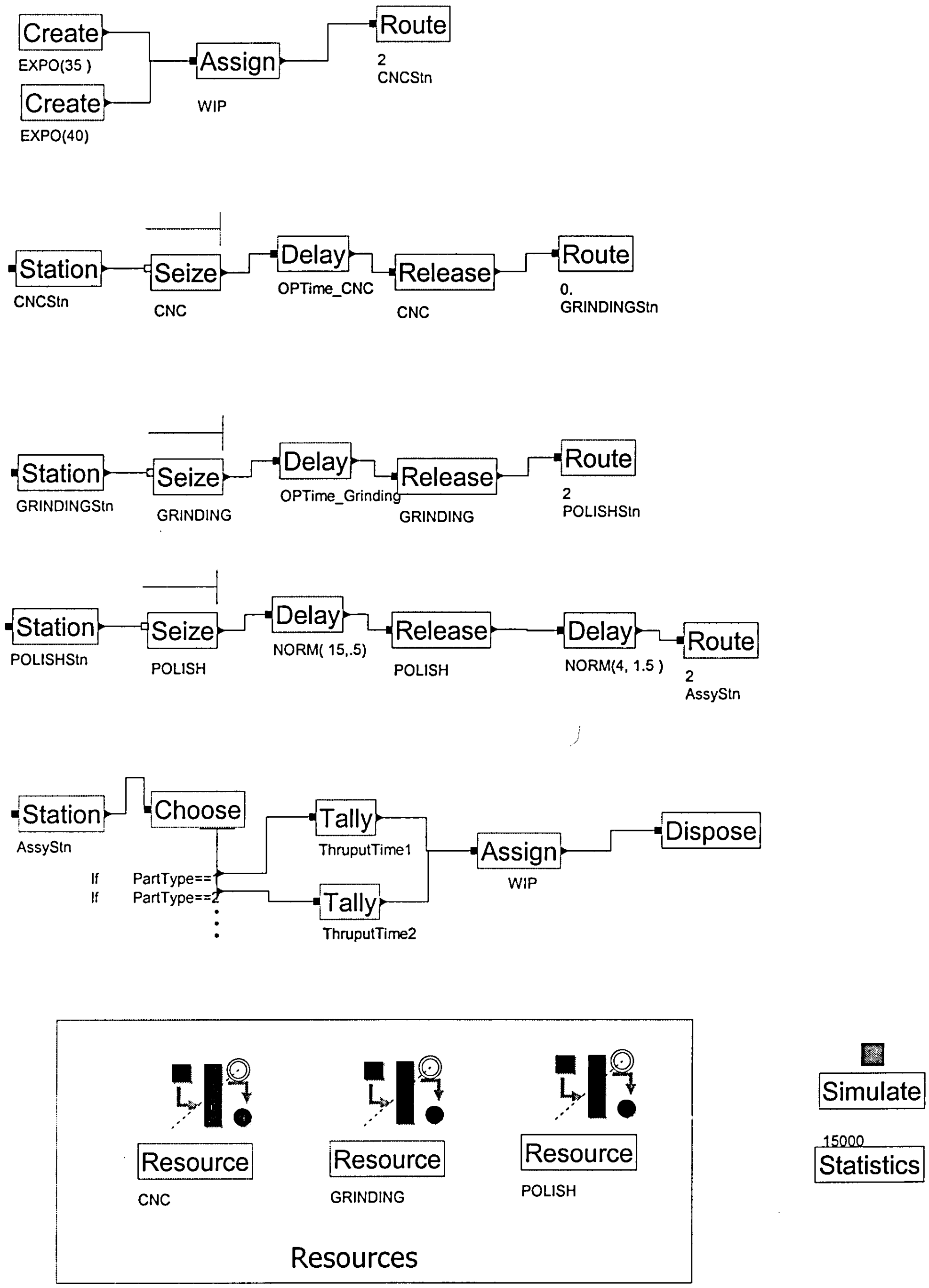


Figure 5.9 Arena simulation model for scenario 1

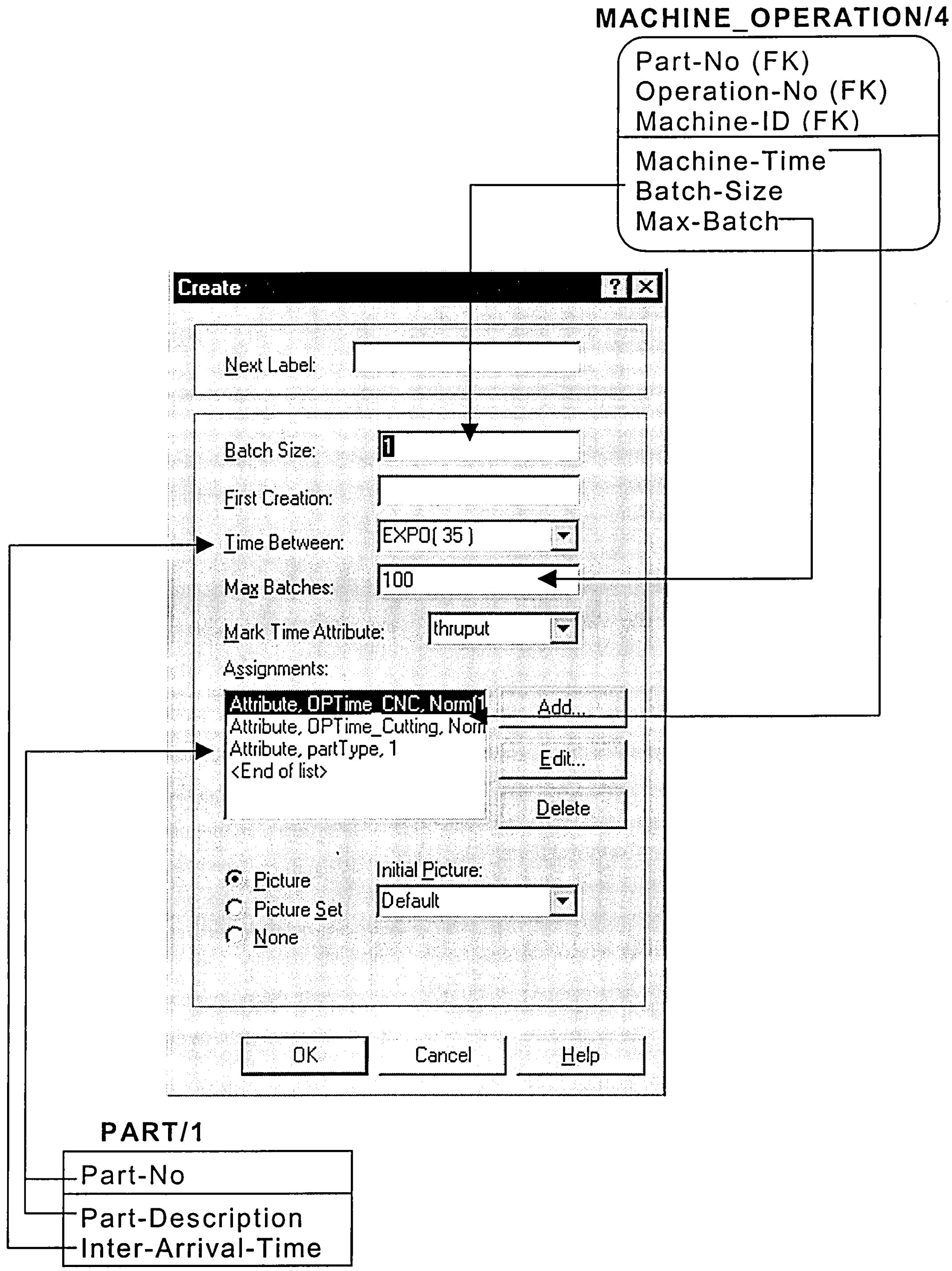


Figure 5.10 Data mapping between entity model and constructed arena model for scenario 1

Factor	Average	Minimum	Maximum
Throughput Time (minutes)			
• Part 1	3295.3	1262.8	5588.7
• Part 2	3383.8	1240.1	5653.1
Total Production (parts)			
• Part 1	219	-	-
• Part 2	209	-	-
Resources Utilisation (%)			
• CNC Lathe	100	100	100
• Grinding	100	100	100
• Polish	52.67	0	100
Queue Size			
• Queue at CNC	137	42	230
• Queue at Grinding	47	16	81
• Queue at Polish	0	0	0

Table 5.4. Sample simulation outputs for scenario 1

Scenario 2

Level of detail is medium

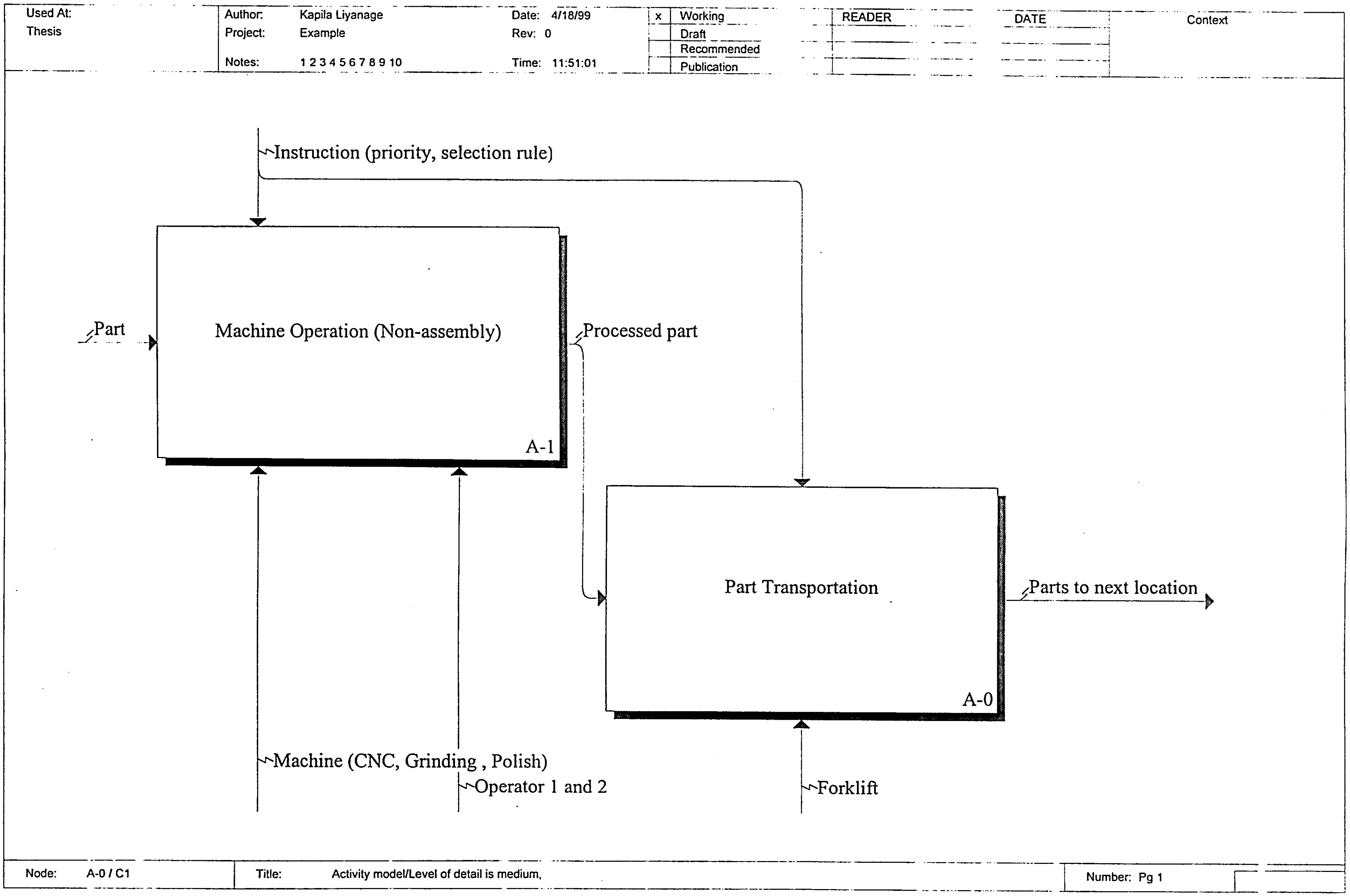


Figure 5.11 Activity model for scenario 2

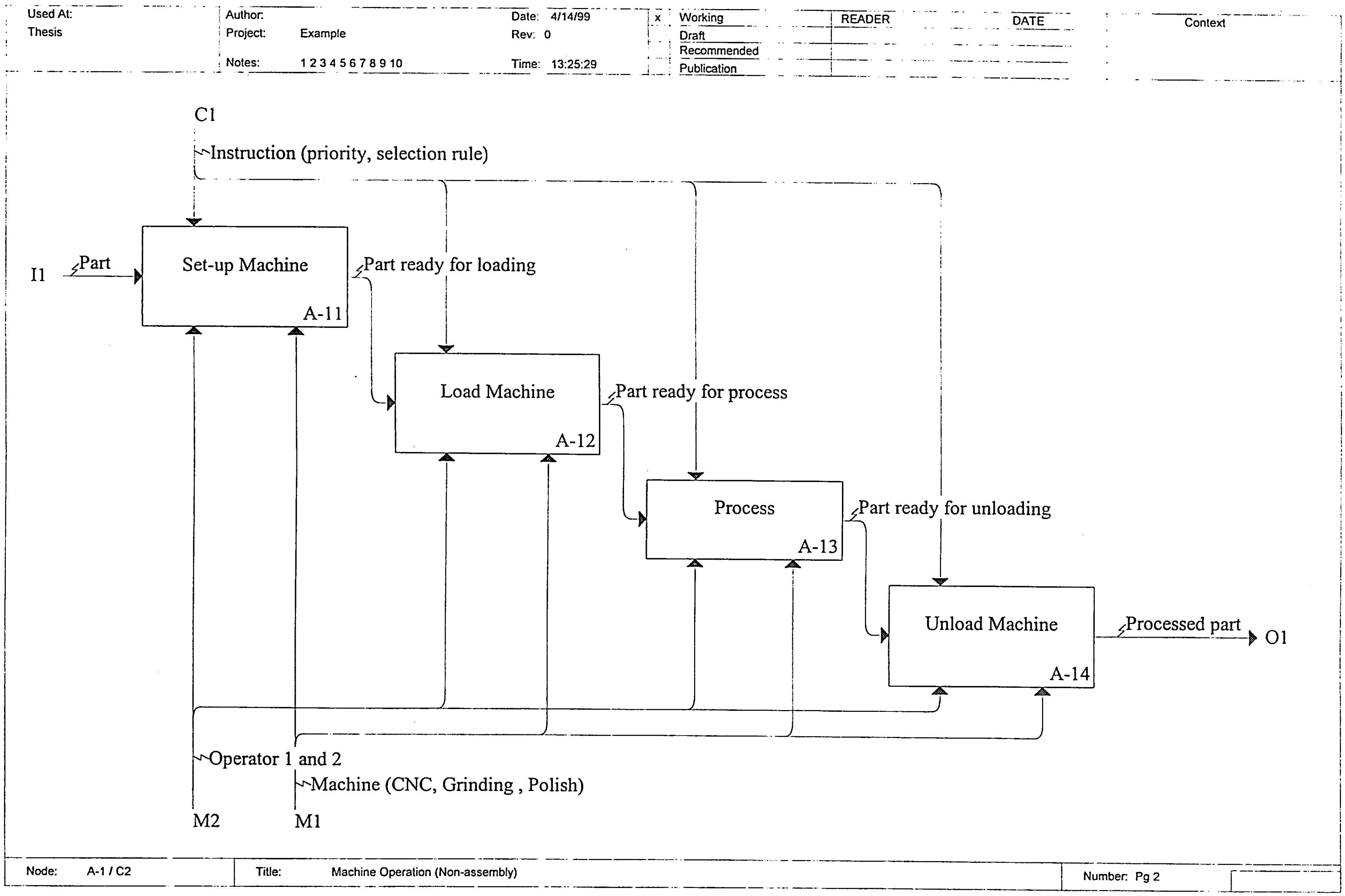


Figure 5.11 (continued) Activity model for scenario 2

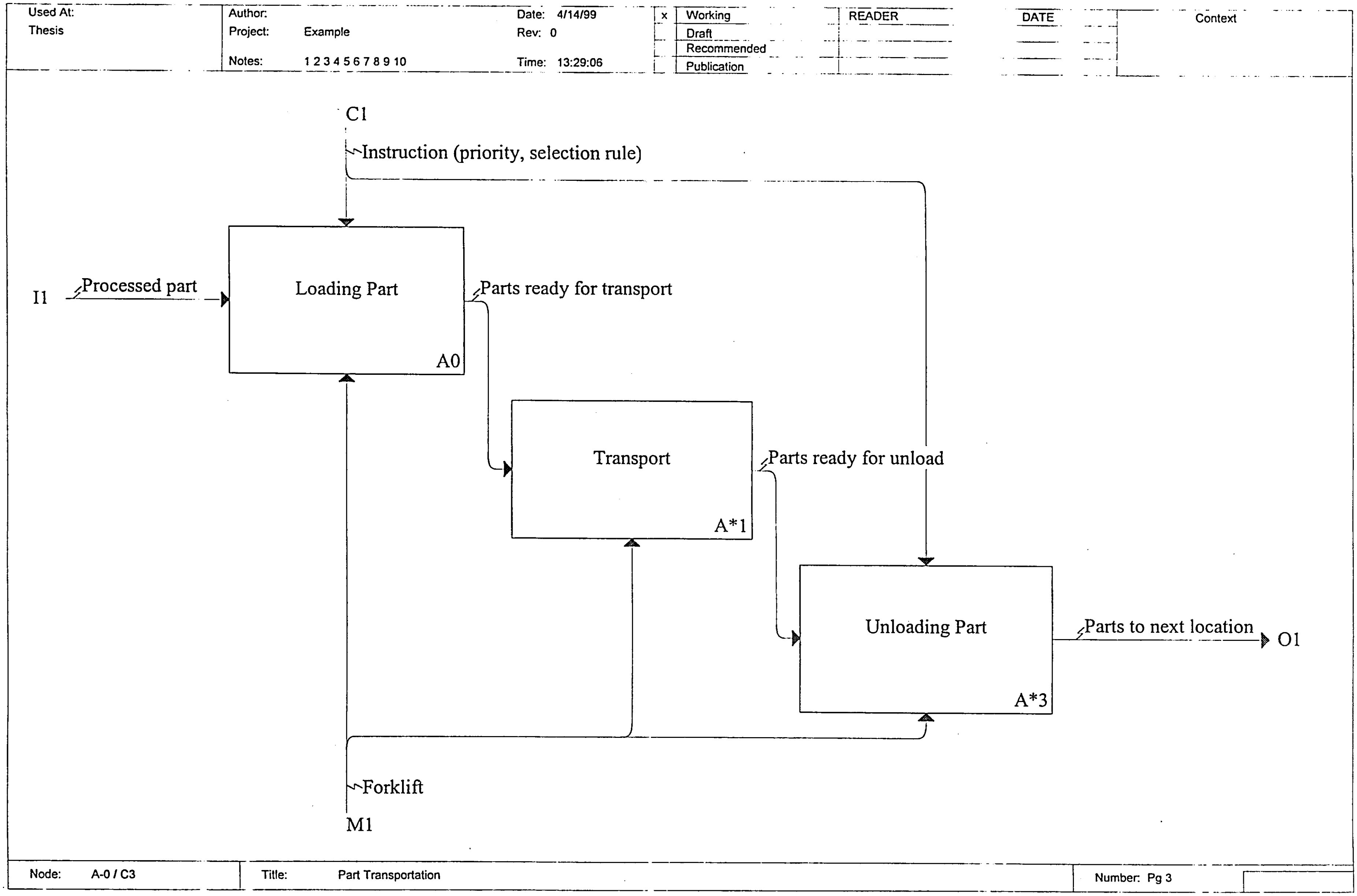


Figure 5.11 (continued) Activity model for scenario 2

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part (I)	Part ID	PART	CREATE
		Part Description	PART	CREATE
		Batch Size	MACHINE_OPERATION	CREATE
		Max Batches	MACHINE_OPERATION	CREATE
		Inter Arrival Time	MACHINE_OPERATION	CREATE
	Machine (M)	Machine ID	MACHINE	RESOURCE/SEIZE
		Machine Desciption	MACHINE	RESOURCE/SEIZE
		MTBF	MACHINE_GROUP	RESOURCE
		MTTR	MACHINE_GROUP	RESOURCE
		Input Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
		Output Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
	Operator (M)	Operator ID	OPERATOR	RESOURCE/SEIZE
		Operator Description	OPERATOR	RESOURCE/SEIZE
		Efficency	OPERATOR	
		Skills	OPERATOR	
		Learning Curve Effect	OPERATOR	
	Machine Operation (Non-assembly) (F)	Machine Operation Desc.	MACHINE_OPERATION	
		Machine Time	MACHINE_OPERATION	DELAY
		Operator Responsibilities	MACHINE_OPERATION	
	Schedule (C)	Schedule Name	SCHEDULE	RESOURCE-->Schedule block
		Schedule Description	SCHEDULE	RESOURCE-->Schedule block
		Schedule Time	SCHEDULE	RESOURCE-->Schedule block
		Duration	SCHEDULE	RESOURCE-->Schedule block
		Machine available time	MACHINE_SHEDULE	RESOURCE-->Schedule block
		Machine unavailable time	MACHINE_SHEDULE	RESOURCE-->Schedule block
		Operator available time	OPERATOR_SHEDULE	RESOURCE-->Schedule block
		Operator unavailable time	OPERATOR_SHEDULE	RESOURCE-->Schedule block
	Instruction (C)	Priority Rule	MACHINE_OPERATION	SEIZE

Table 5.5 (a) Mapping table for scenario 2 [machine process module (Non-assembly)]

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
2	Described in Levels 1 +	Described in Levels 1 & 2		
	Set-up Machine **(F)**	Set-up Description	MACHINE_OPERATION	
		Set-up Time	MACHINE_OPERATION	DELAY/VARIABLE
	Load Machine **(F)**	Loading Time	MACHINE_OPERATION	DELAY
	Unload Machine **(F)**	Unloading Time	MACHINE_OPERATION	DELAY

Table 5.5 (a) (continued) Mapping table for scenario 2 [machine process module (Non-assembly)]

Level of Decomposition	Activity_Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part (I)	Part ID	PART	CREATE
		Part Description	PART	CREATE
	Part Transportation (F)	Beginning Station	TRANSFER_OPERATION	TRANSPORT or SEGMENT or ROUTE
		Number of parts to Load at this Stn	TRANSFER_OPERATION	TRANSPORT
		Ending Station	TRANSFER_OPERATION	TRANSPORT
		Number of parts to Load at this Stn	TRANSFER_OPERATION	TRANSPORT or SEGMENT or ROUTE
		Distance	TRANSFER_OPERATION	DISTANCE
		Travel time	TRANSFER_OPERATION	ROUTE
		Operator Responsibilities	TRANSFER_OPERATION	
	Transporter (M)	Device ID	MH_DEVICE	TRANSPORTER
		Velocity	MH_DEVICE	TRANSPORTER
		MTBF	MH_DEVICE	TRANSPORTER
		MTTR	MH_DEVICE	TRANSPORTER
		Number of Transporter units	MH_DEVICE	TRANSPORTER
	Operator (M)	Operator ID	OPERATOR	RESOURCE
		Operator Description	OPERATOR	RESOURCE
		Efficiency		
		Skills		
		Learning Curve Effect		
2	Instruction (C)	Priority	TRANSFER_OPERATION	ALLOCATE
	Described in Levels 1 +	Described in Levels 1 +		
	Loading Part (F)	Loading Time	TRANSFER_OPERATION	DELAY
	Unloading Part (F)	Unloading Time	TRANSFER_OPERATION	DELAY

Table 5.5 (b) Mapping table for scenario 2 [material handling module]

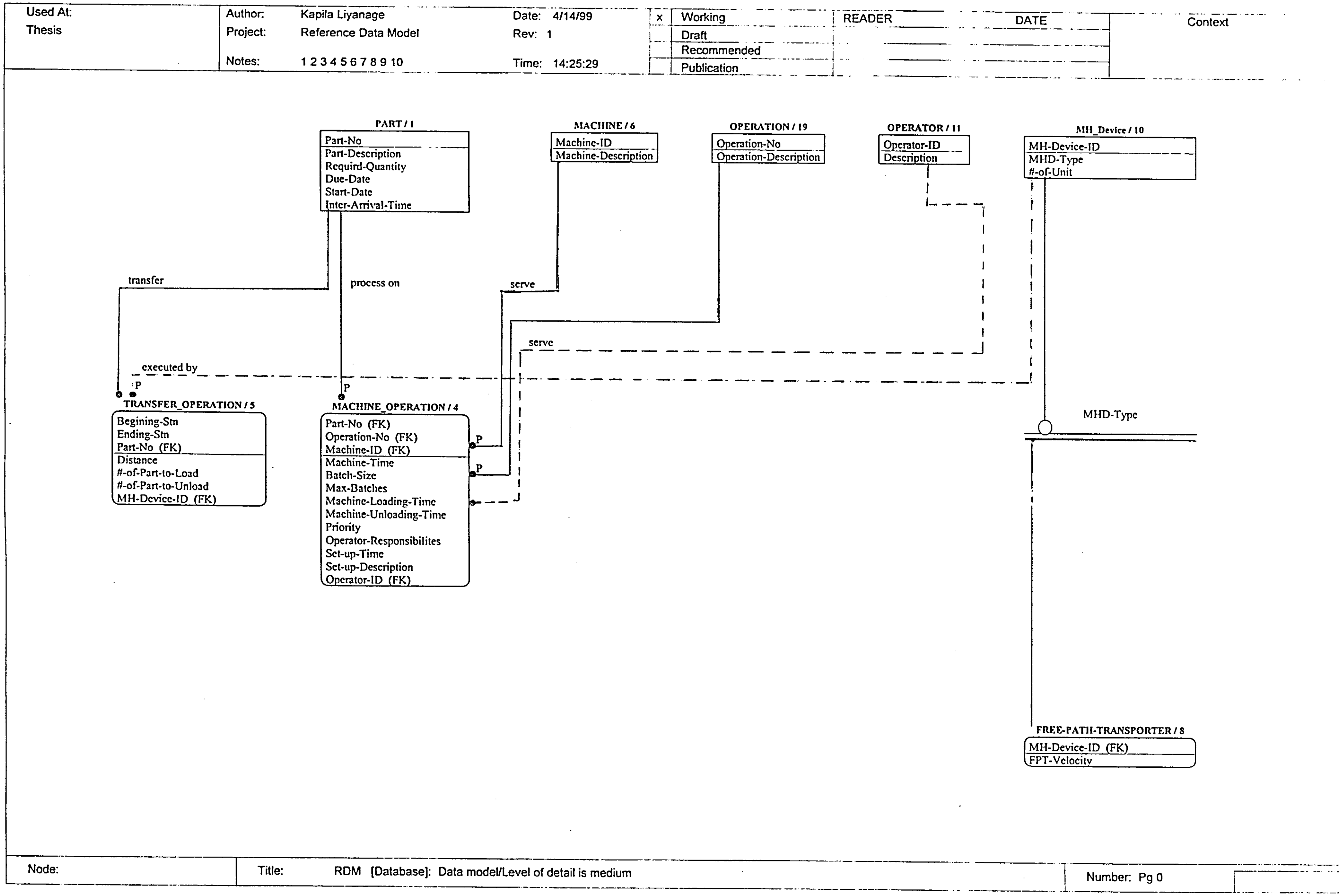


Figure 5.12 The required entity model for scenario 2

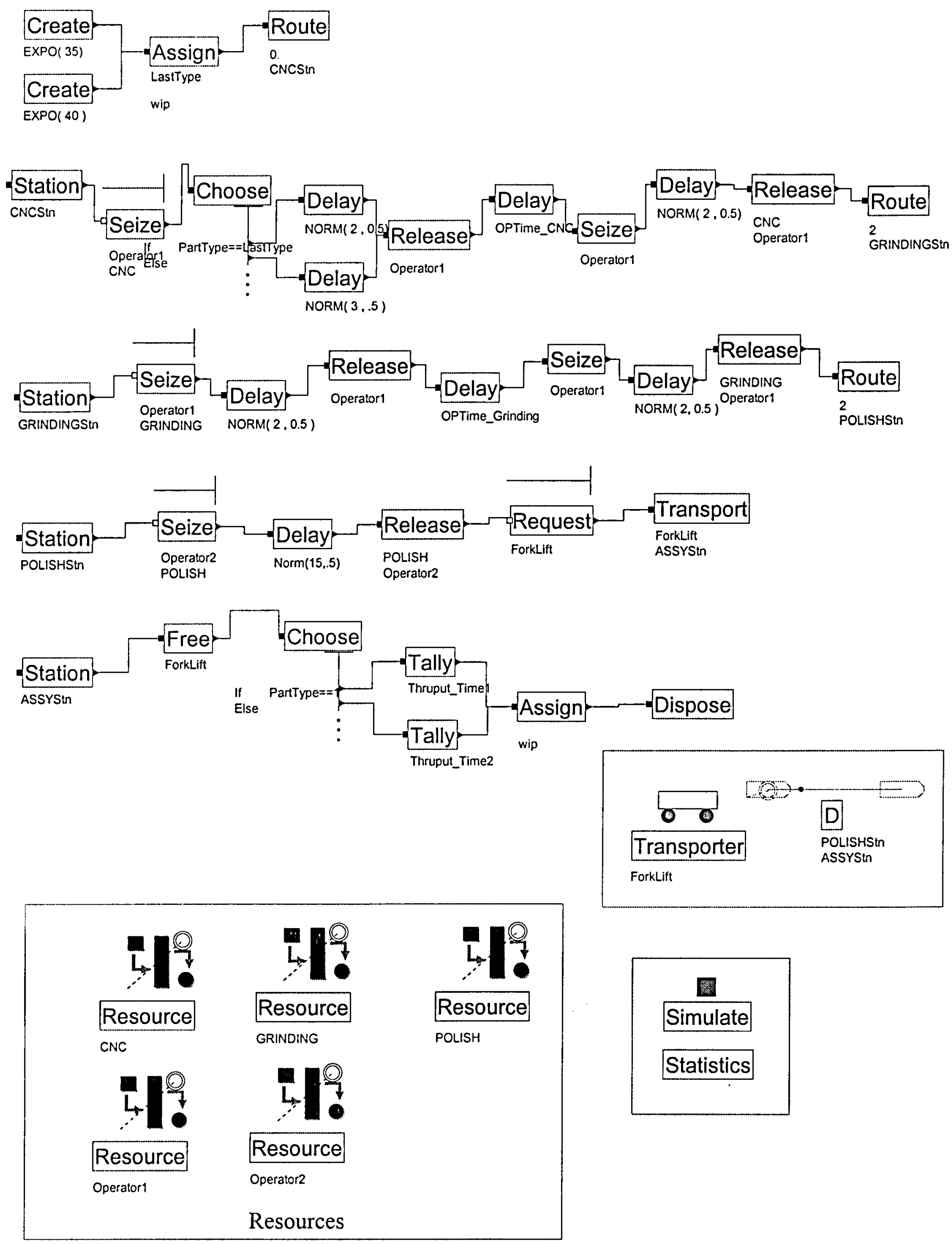


Figure 5.13 Arena simulation model for scenario 2

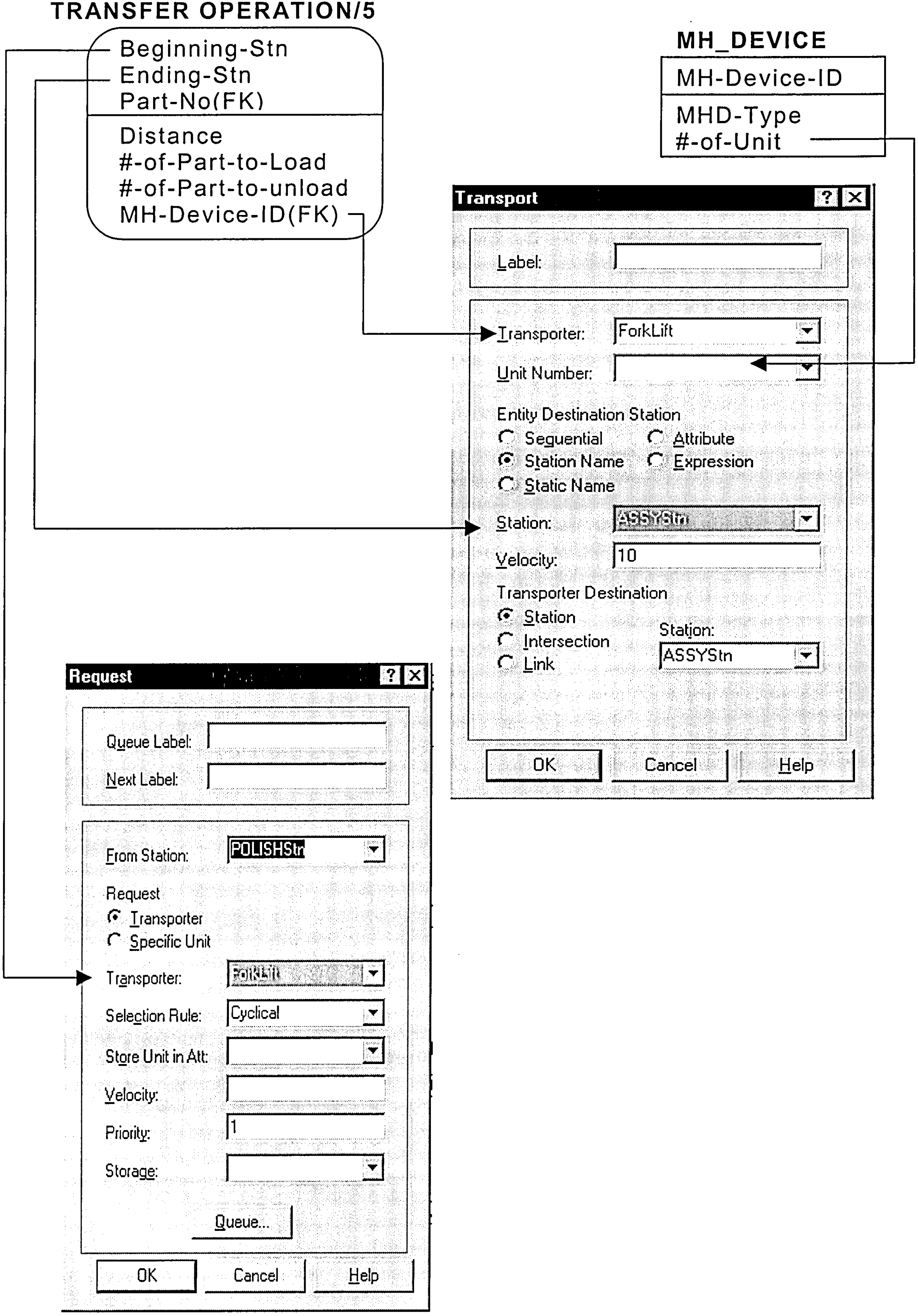


Figure 5.14 Data mapping between entity model and constructed arena model for scenario 2

Factor	Average	Minimum	Maximum
Throughput Time (minutes)			
• Part 1	1954.6	466.29	3737.1
• Part 2	1943.4	485.55	3736.9
Total Production (parts)			
• Part 1	277	-	-
• Part 2	221	-	-
Resources Utilisation (%)			
• CNC Lathe	100	100	100
• Grinding	100	100	100
• Polish	62.47	0	100
• Operator 1	37.39	0	100
• Operator 2	62.45	0	100
• Forklift	16.6	0	100
Queue Size			
• Queue at CNC	104	21	193
• Queue at Grinding	2	0	6
• Queue at Polish	0	0	1
• Queue at Forklift	0	0	0

Table 5.6 Sample simulation outputs for scenario 2

Scenario 3

Level of detail is high

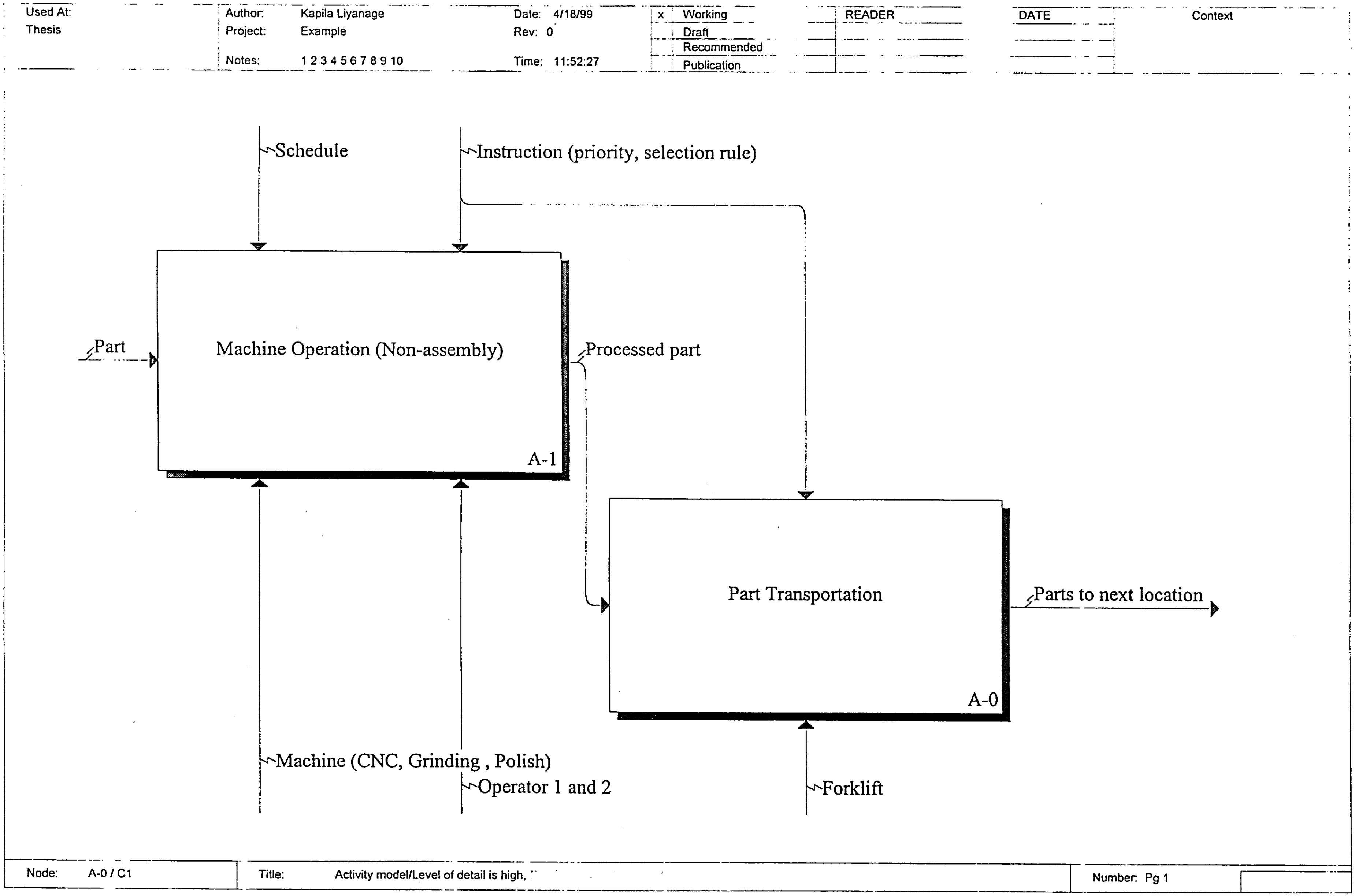


Figure 5.15 Activity model for scenario 3

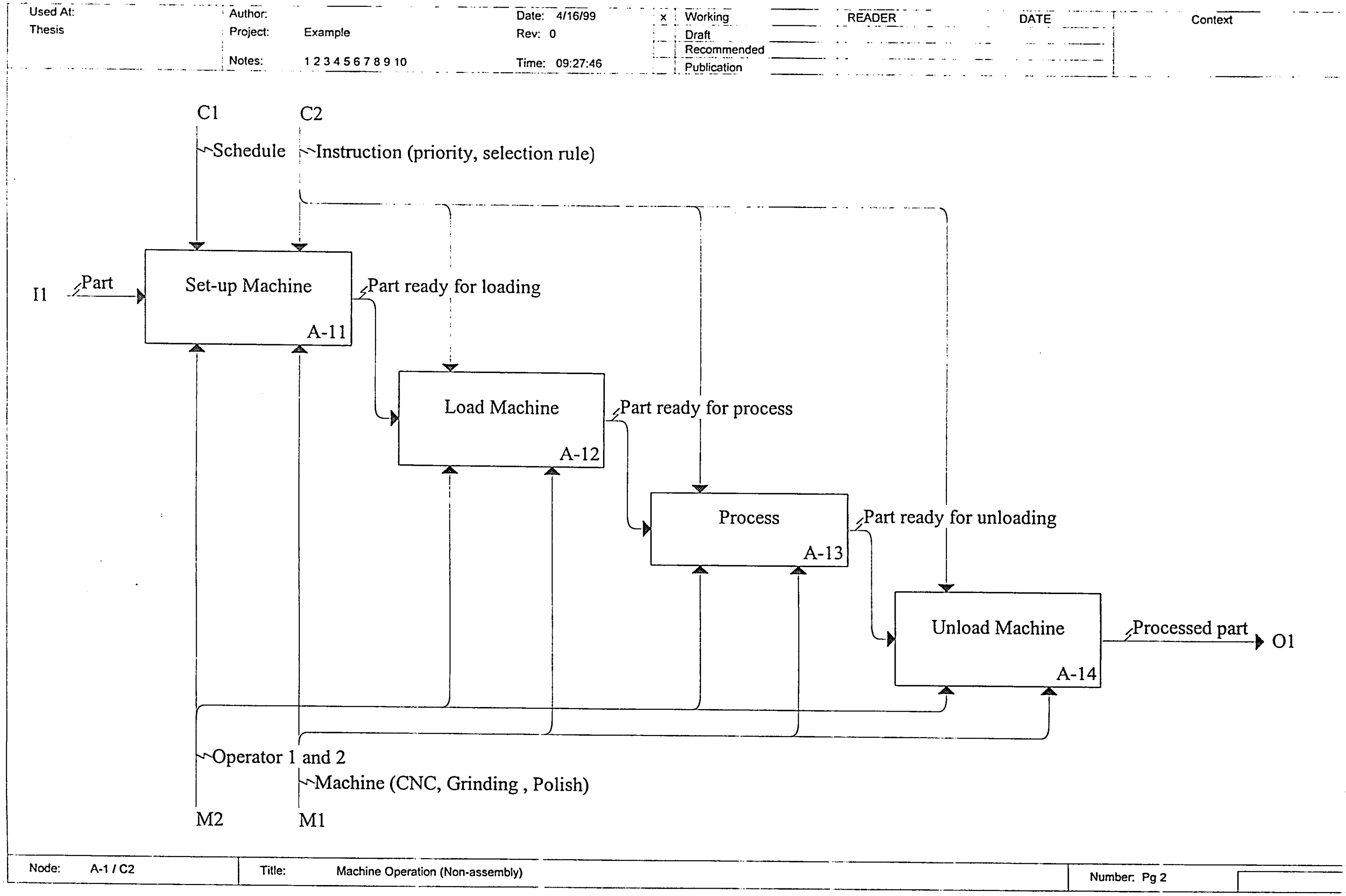


Figure 5.15 (continued) Activity model for scenario 3

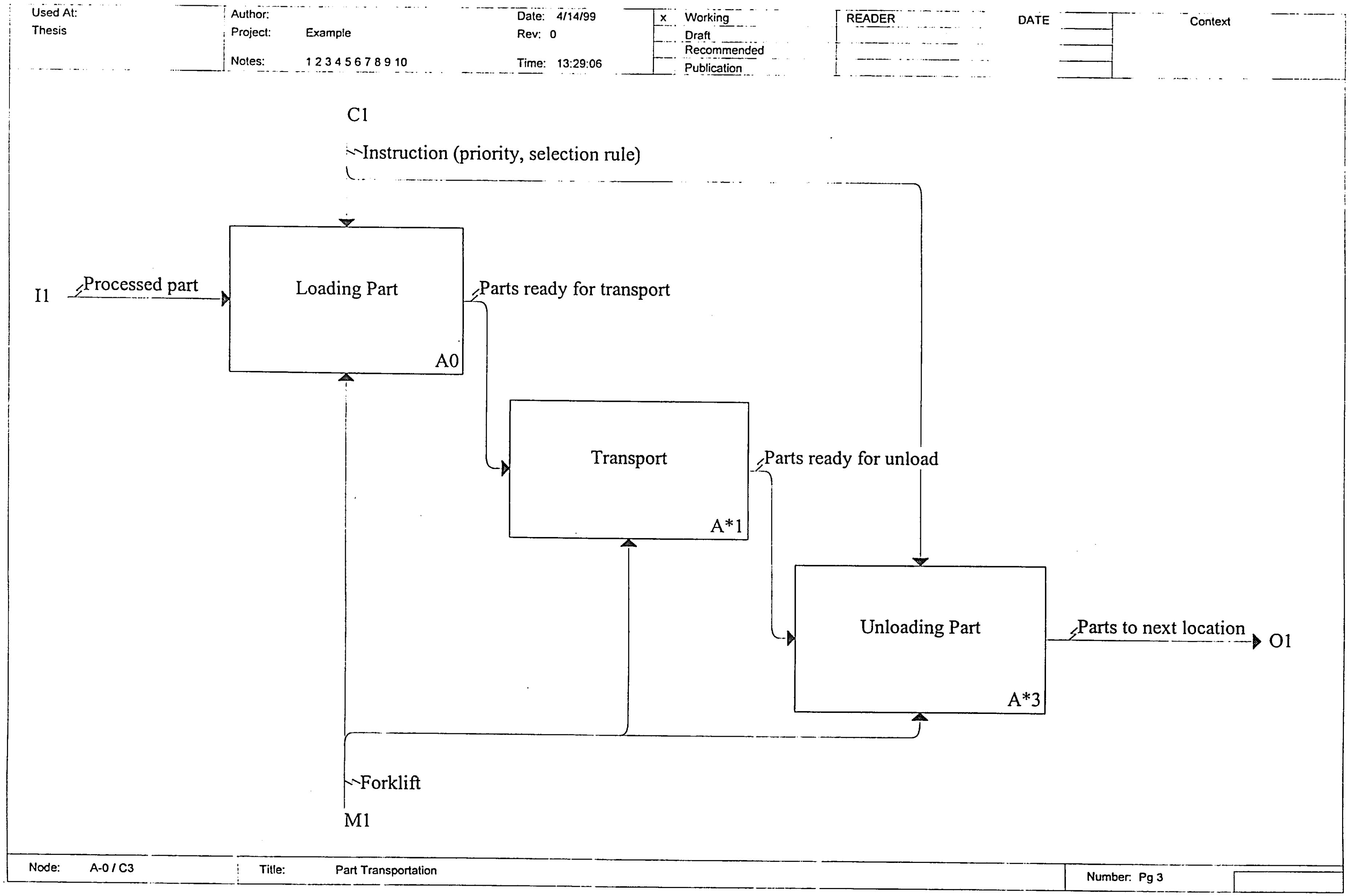


Figure 5.15 (continued) Activity model for scenario 3

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part **(I)**	Part ID	PART	CREATE
		Part Description	PART	CREATE
		Batch Size	MACHINE_OPERATION	CREATE
		Max Batches	MACHINE_OPERATION	CREATE
		Inter Arrival Time	MACHINE_OPERATION	CREATE
	Machine **(M)**	Machine ID	MACHINE	RESOURCE/SEIZE
		Machine Description	MACHINE	RESOURCE/SEIZE
		MTBF	MACHINE_GROUP	RESOURCE
		MTTR	MACHINE_GROUP	RESOURCE
		Input Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
		Output Buffer Capacity	MACHINE_GROUP	RESOURCE or STORAGE
	Operator **(M)**	Operator ID	OPERATOR	RESOURCE/SEIZE
		Operator Description	OPERATOR	RESOURCE/SEIZE
		Efficiency	OPERATOR	
		Skills	OPERATOR	
		Learning Curve Effect	OPERATOR	
	Machine Operation (Non-assembly) **(F)**	Machine Operation Desc.	MACHINE_OPERATION	
		Machine Time	MACHINE_OPERATION	DELAY
		Operator Responsibilities	MACHINE_OPERATION	
	Schedule **(C)**	Schedule Name	SCHEDULE	RESOURCE-->Schedule block
		Schedule Description	SCHEDULE	RESOURCE-->Schedule block
		Schedule Time	SCHEDULE	RESOURCE-->Schedule block
		Duration	SCHEDULE	RESOURCE-->Schedule block
		Machine available time	MACHINE_SCHEDULE	RESOURCE-->Schedule block
		Machine unavailable time	MACHINE_SCHEDULE	RESOURCE-->Schedule block
		Operator available time	OPERATOR_SCHEDULE	RESOURCE-->Schedule block
		Operator unavailable time	OPERATOR_SCHEDULE	RESOURCE-->Schedule block
	Instruction **(C)**	Priority Rule	MACHINE_OPERATION	SEIZE

Table 5.7(a) Mapping table for scenario 3 [machine process module (Non-assembly)]

Level of decomposition	Activity Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
2	Described in Levels 1 +	Described in Levels 1 & 2		
	Set-up Machine **(F)**	Set-up Description	MACHINE_OPERATION	
		Set-up Time	MACHINE_OPERATION	DELAY/VARIABLE
	Load Machine **(F)**	Loading Time	MACHINE_OPERATION	DELAY
	Unload Machine **(F)**	Unloading Time	MACHINE_OPERATION	DELAY

Table 5.7(a) (continued) Mapping table for scenario 3 [machine process module (Non-assembly)]

Level of Decomposition	Activity_Modelling Element	Required Data	Reference Model Path (Corresponding RM Entity)	Corresponding ARENA Template
1	Part **(I)**	Part ID	PART	CREATE
		Part Description	PART	CREATE
	Part Transportation **(F)**	Beginning Station	TRANSFER_OPERATION	TRANSPORT or SEGMENT or ROUTE
		Number of parts to Load at this Stn	TRANSFER_OPERATION	TRANSPORT
		Ending Station	TRANSFER_OPERATION	TRANSPORT
		Number of parts to Load at this Stn	TRANSFER_OPERATION	TRANSPORT or SEGMENT or ROUTE
		Distance	TRANSFER_OPERATION	DISTANCE
		Travel time	TRANSFER_OPERATION	ROUTE
		Operator Responsibilities	TRANSFER_OPERATION	
	Transporter **(M)**	Device ID	MH_DEVICE	TRANSPORTER
		Velocity	MH_DEVICE	TRANSPORTER
		MTBF	MH_DEVICE	TRANSPORTER
		MTTR	MH_DEVICE	TRANSPORTER
		Number of Transporter units	MH_DEVICE	TRANSPORTER
	Operator **(M)**	Operator ID	OPERATOR	RESOURCE
		Operator Description	OPERATOR	RESOURCE
		Efficiency		
		Skills		
		Learning Curve Effect		
2	Instruction **(C)**	Priority	TRANSFER_OPERATION	ALLOCATE
	Described in Levels 1 +	Described in Levels 1 +		
	Loading Part **(F)**	Loading Time	TRANSFER_OPERATION	DELAY
	Unloading Part **(F)**	Unloading Time	TRANSFER_OPERATION	DELAY

Table 5.7(b) Mapping table for scenario 3 [material handling module]

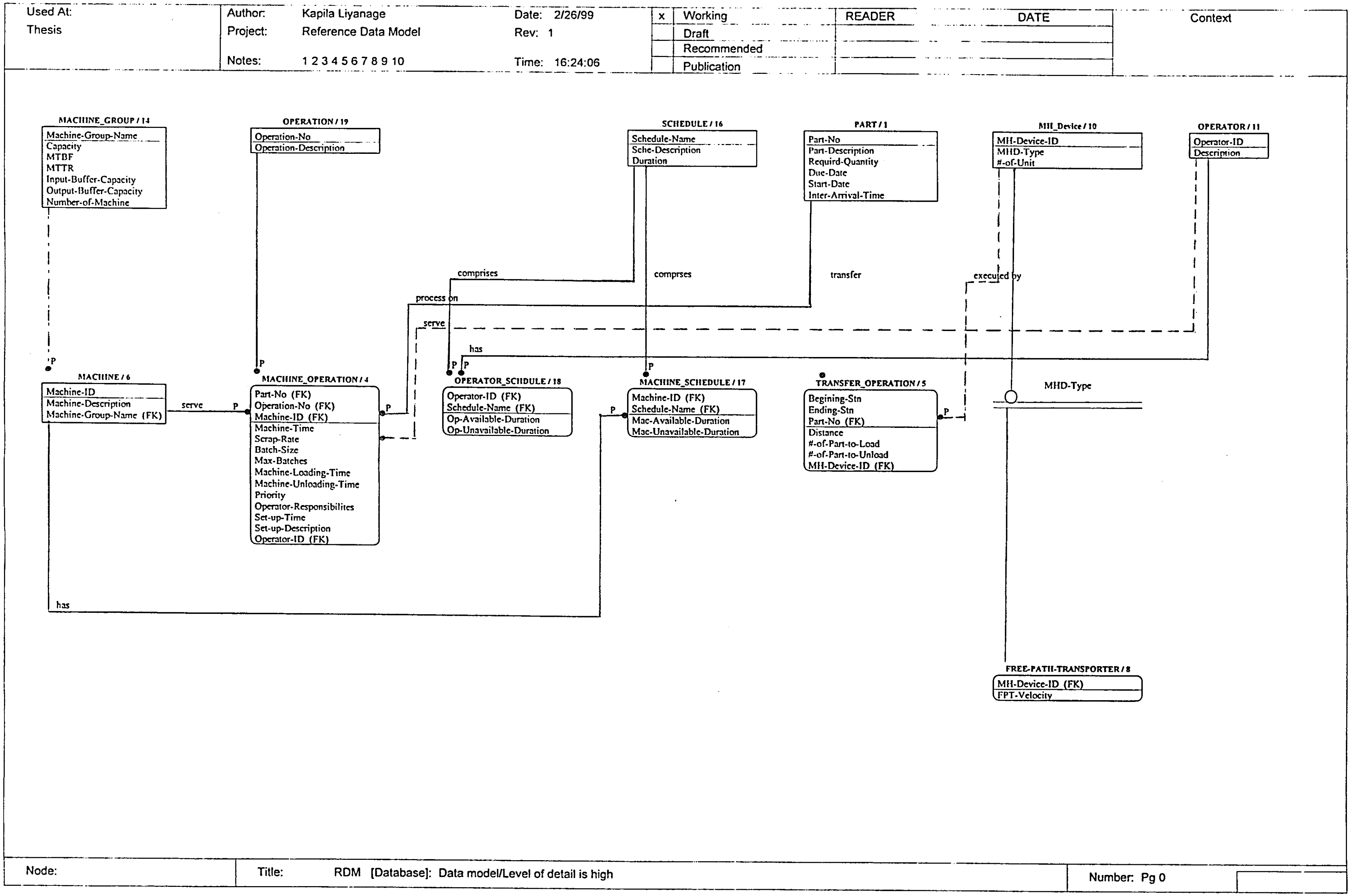
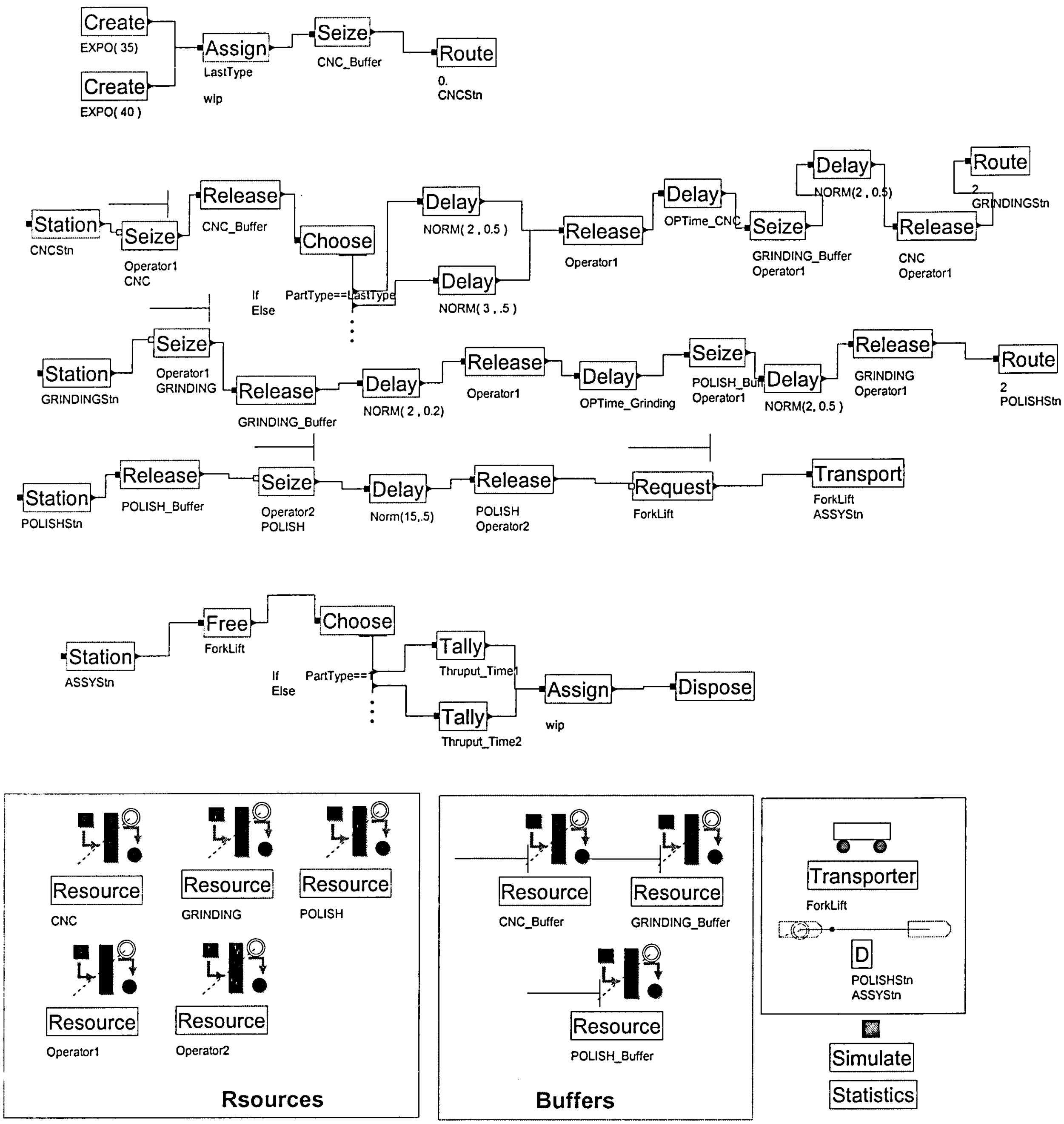


Figure 5.16 The required entity model for scenario 3



(Schedules and breakdowns have been modelled inside the resources blockes)

Figure 5.17 Arean simulation for scenario 3

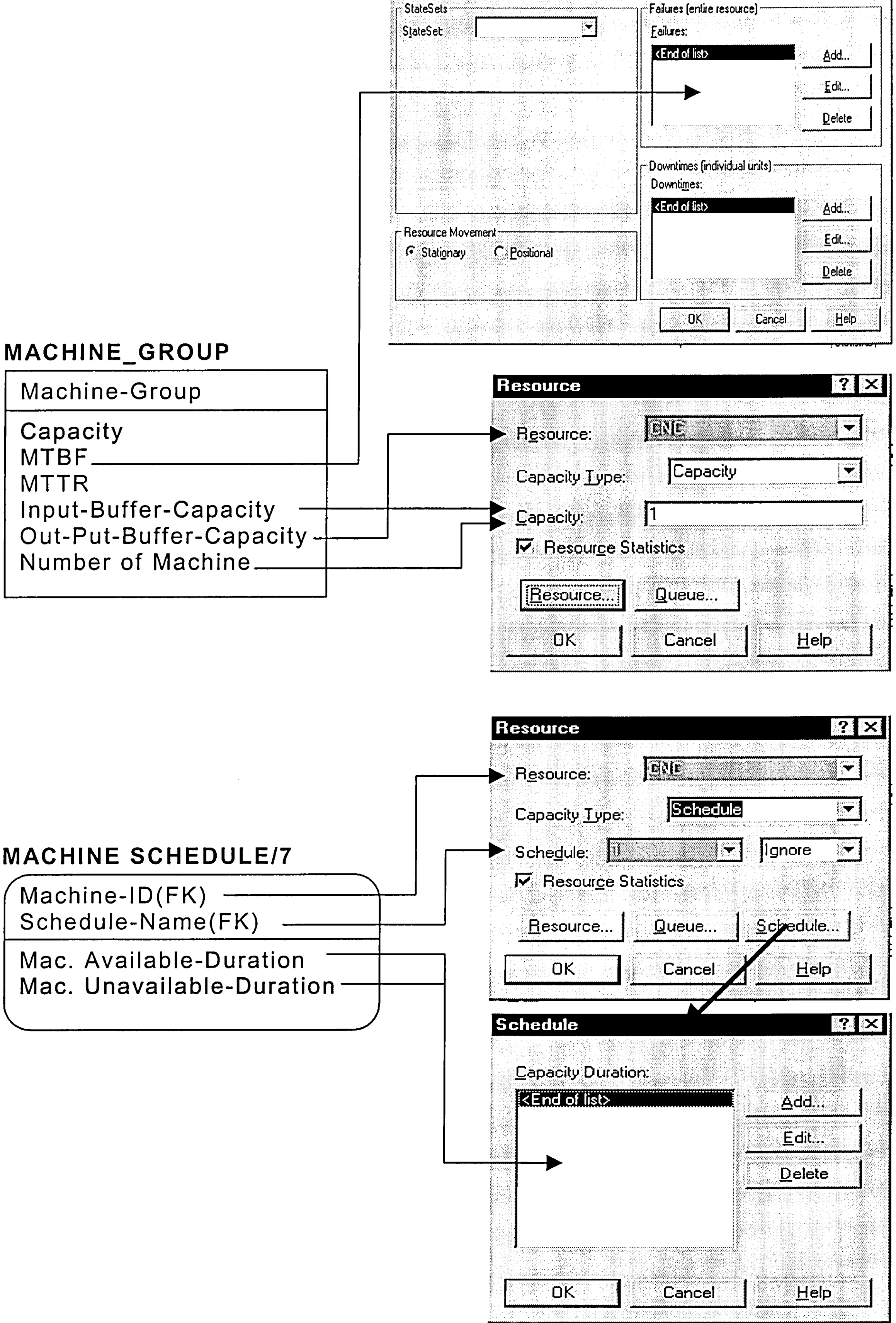


Figure 5.18 Data mapping between entity model and constructed arena model for scenario 3

Factor	Average	Minimum	Maximum
Throughput Time (minutes)			
• Part 1	2668.2	1060.7	4556.1
• Part 2	2675.7	1074.1	4582.9
Total Production (parts)			
• Part 1	233	-	-
• Part 2	226	-	-
Resources Utilisation (%)			
• CNC Lathe	97.85	0	100
• Grinding	95.36	0	100
• Polish	57.36	0	100
• Operator 1	34.53	0	100
• Operator 2	57.36	0	100
• Forklift	15.30	0	100
Queue Size			
• Queue at CNC	1	1	1
• Queue at Grinding	0	0	1
• Queue at Polish	0	0	1
• Queue at Forklift	0	0	0
Resources Availability (%)			
• CNC	97.05	0	100
• Grinding	98.25	0	100
• Polish	98.20	0	100
• Operator 1	99.16	0	100
• Operator 2	99.16	0	100
• Forklift	100	100	100
Resources Downtimes (%)			
• CNC	2.10	-	-
• Grinding	1.11		
• Polish	0.95		

Table 5.8 Sample simulation outputs for scenario 3

5.4 Other Advantages of the Methodology

This methodology has several other merits. The proposed methodology (MMOD) can also support other stages of a simulation project, namely, the model representation stage, data validation and verification, etc.

5.4.1 Model Representation

Model representation is the process of translating the conceptual model into a communicative model which can be communicated to another person to understand the behaviour of the system (Balci, 1990).

There are many diagrammatic modelling techniques available for model representation [Schorman & Perera (1998) and Ceric & Paul (1992)], namely, flow chart, IDEF, activity cycle diagrams, extended activity cycle diagrams, hierarchical activity cycle diagrams and petri-nets, etc.

For example, both the IDEF0 diagram and Activity Cycle Diagram (ADC) are applied to illustrate the activities concerning machining operation. The activity cycle diagram is used to compare IDEF0 diagrams since the activity cycle diagram (ACD) is a popular diagrammatic technique which is specifically used in simulation (Pidd, 1992). In this example, the production cell contains a machine and a operator. The activities used in machining operation are *load part*, *process part* and *unload part.*

IDEF0 modelling diagram

The diagram shown in figure 5.19 illustrates the decomposition of the activity 'part-processing'. Its Input is part, Control is instruction, Output is processed part and the Mechanisms are the machine and operator. The next decomposition layer consists of activity 'Load machine', 'Process', and 'Unload machine'. The knowledge from IDEF0 diagrams attempt to express the logic of a system to be simulated as follows.

1. Start loading when a part is available and a operator available at machine. (instruction describes which part to be selected to load)

2. Operator switch on the machine and machine is running following instruction.

3. When a machining operation is completed, the part will then be unloaded from machine.

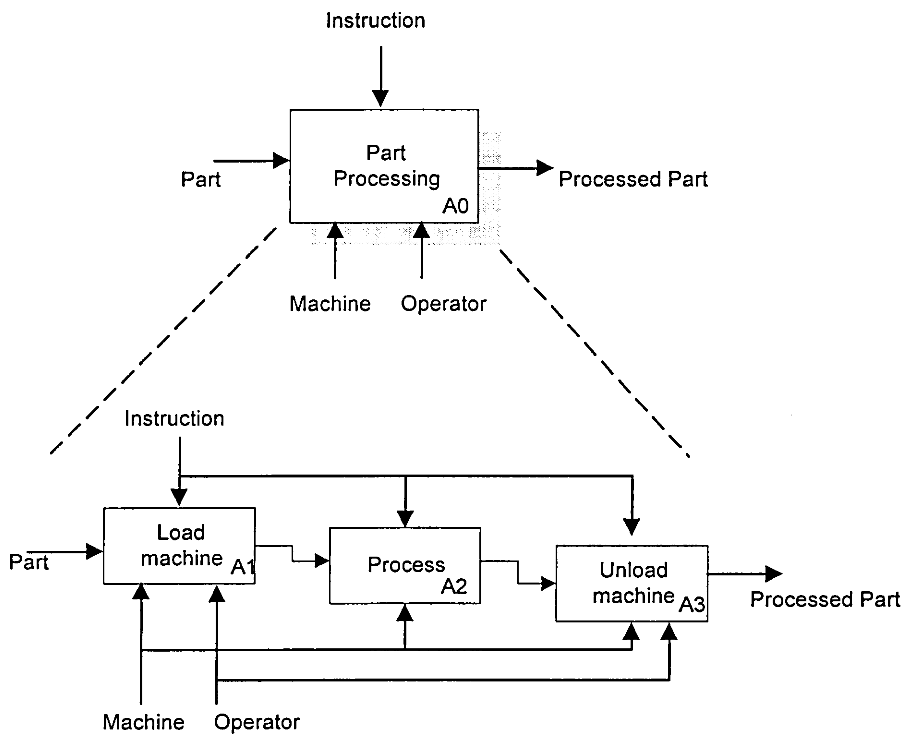


Figure 5.19 IDEF0 activity diagram representation

Activity Cycle Diagram (ACD)

An ACD provides a means of describing the logic of a simulation model. This diagrammatic technique is capable of understanding the underlying logic of the problem (Au and Paul, 1997). Activity cycle diagrams consist of only two symbols: a circle to represent the idle state of a resource and a rectangle box representing an active state.

The diagram shown in figure 5.20 illustrates the interaction between a machine, parts and a operator. During this simple machine operation, a part enters the system and waits in a queue for loading. The operator is required to load, unload and set-up the machine. The activity "loading" and "unloading" represents the first and last phase of the diagram. When the machine is ready for loading and an operator is available to load the machine, the next step is to load the machine. After the part is loaded into the machine, the machine starts to process the part. The activity 'process' represents the middle phase of the diagram. When a machining operation is completed, the part will then be unloaded from the machine (under condition of availability of operator) and the system is ready for another operation.

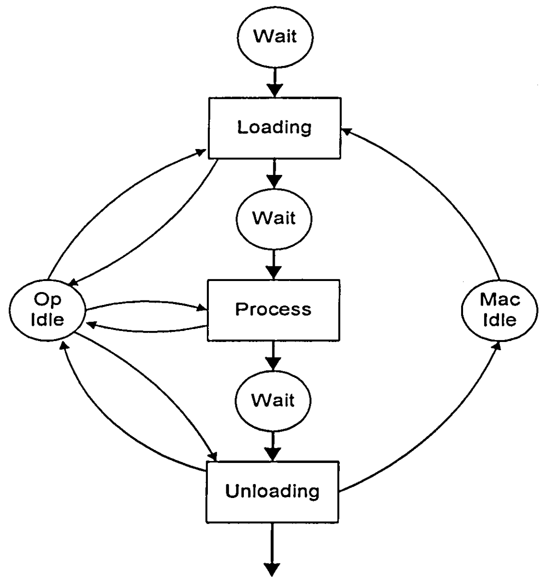


Figure 5.20 Activity cycle diagram representation

By comparing the above two diagrams, the IDEF0 activity model library can also be used for the model representation stage at a certain limit. The disadvantage of IDEF0 is that it can only represent the static behaviour of a system. It does is not provide a structure to describe simulation complexity. However, most of the facts can be illustrated by knowledge from inputs, control, output and the mechanism (ICOMs) of IDEF0. The main advantage of the ACD is that it provides an insight into the problem and understanding of the basic structure and the logic of the system being modelled, specifically from a simulation point of view. However, according to the Schorman's and Perera's (1998), most currently applied diagramming techniques for model representation is IDEF (IDEF0 & IDEF3). Possible reasons may be that the IDEF0 is easy to use and it has a hierarchical modelling capability. Activity cycle diagrams are more difficult to use and incapable of handling the full complexity of a real manufacturing system. The activity cycle diagram also provides just a static view of a system, but it shows idle, active states of resources with interaction between them which can not be modelled in IDEF0.

5.4.2 Model building

The relational database, which consists of multiple data tables and user-friendly forms, defines the type of data to be collected. The collated data is conveniently stored in a database which can be accurately maintained and manipulated using SQL-DML according to the user requirements, and can then be directly linked into simulation models as illustrated in figure 5.21.

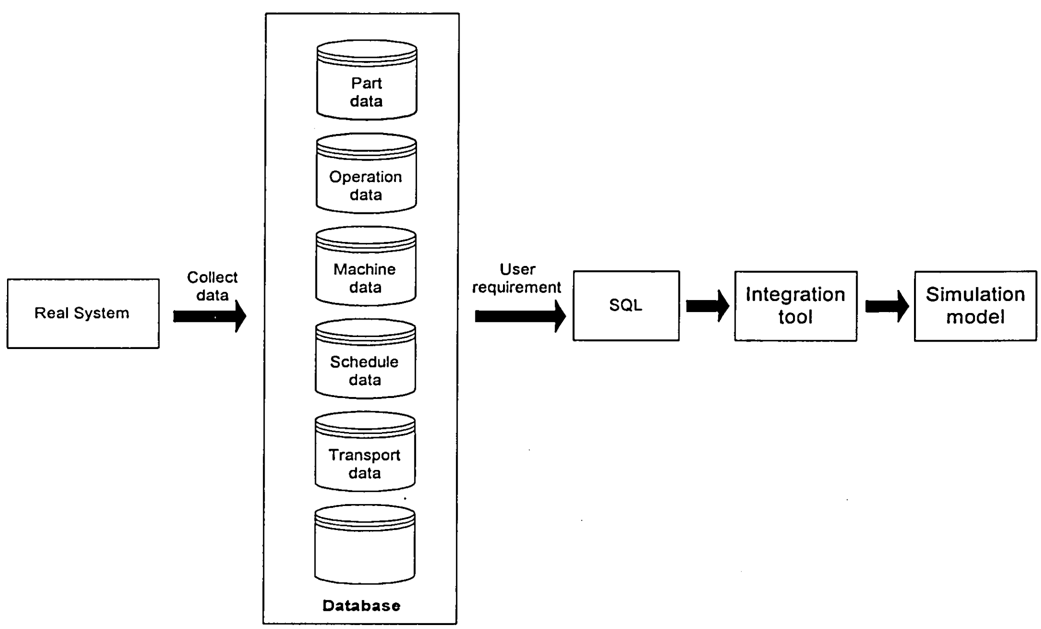


Figure 5.21 Integration approach

For instance, in certain situations the modeller may need to retrieve part routing information as shown in table 5.9.

Part ID					
Operation No	Machine ID	Machine Group	Machine time	Loading time	Unloading time

Table 5.9 Required part routing information

Here is the SQL query used to extract the required part routing information in a MS Access.

```
SELECT DISTINCTROW OPERATION.[Operation No], MACHINE.[Machine ID], MACHINE.[Machine Group
Name], MACHINE_OPERATION_PART.[Machine Time], MACHINE_OPERATION_PART.[Machine Loading
Time], MACHINE_OPERATION_PART.[Machine Loading Time]
FROM MACHINE INNER JOIN (OPERATION INNER JOIN (PART INNER JOIN
MACHINE_OPERATION_PART ON PART.[Part No] = MACHINE_OPERATION_PART.[Part No]) ON
OPERATION.[Operation No] = MACHINE_OPERATION_PART.[Operation No]) ON MACHINE.[Machine ID] =
MACHINE_OPERATION_PART.[Machine ID]
WHERE ((PART.[Part No]=[Enter Part Number]))
ORDER BY OPERATION.[Operation No];
```

In an appropriate operating environment, it is possible to link retrieved data into simulation models via recent advanced integration tools such as OLE (Object Linking and Embedding), ODBC(Open Database Connectivity) and VBA(Visual Basic Application). For example, data describing a certain system contained in Microsoft Excel or Microsoft Access can be directly imported to the Arena using Visual Basic Application (Kelton el al ,1998) or Microsoft Visual C++ (Peters et al 1996).

5.4.3 Verification and Validation

As Robinsion (1994) suggests, literature surveys show a number of alternative issues regarding the purpose and definition of verification and validation.

The purpose of model verification is to ensure that the model has been coded correctly. Model validation is the process of ensuring that a simulation model accurately reproduces the behaviour of the real system. The use of different methods between verification and validation are summarised in table 5.10.

Methods of verification	Methods of validation
• Checking the code and logic • Visual checking • Inspecting output reports	• Face validity • Comparison with the real system • Comparison with other models

Table 5.10 Model verification and validation (Based on Robinson, 1994)

According to the literature on model verification and validation, it is important to note that data verification and data validation overlap each other. During the verification period, the modeller needs to check the code to ensure that the right data and logic have been entered, whilst during the validation period, the model data needs to be compared with real world data.

Based on the above discussion, it is obvious that the purpose of data validation is to determine the accuracy of the data that is being used. Balichi (1990) identified the following indicators to measure data validity.

1. Does each input data model process a sufficiently accurate representation?
2. Are the parameter values identified, measured, or estimated with sufficient accuracy?
3. How reliable are the instruments used for data collection and measurement?
4. Are all data transformations done accurately ?
5. Is the dependence between the input variables, if any, represented by the input data model(s) with sufficient accuracy?
6. Is all data up-to-date?

Incorrectly entered data into the model can have a significant impact on simulation experimental results. These types of data errors can happen by;

- directly inputting invalid or wrong data into the simulation model
- recording invalid or wrong data on the spreadsheet, database and even in paper based documents

Incorrectly entered data can be very difficult to detect with a simulation model. A single data entry error, such as a misplaced decimal point or an extra digit in a number can have a significant impact on the experimental result (Pegden et al, 1990). These type of simple errors can happen in almost all simulation projects. Ideally, the best way to cope with these errors is to prevent inputting invalid or wrong data into the model by the introduction of some automated validation rules before entering data, rather than trying to deal with them afterwards. Thus, there is a need to partially automate the validation and verification of input data to maximise the accuracy of the information in the database. Some of the validation rules which can be used in developed MS Access databases are discussed here.

- Format definition

  Format property is used to control how the data elements appear on screen. The modeller can include formats properties as he or she wants. i.e. it is possible to include literal characters (Such as hyphens) and they automatically appear in the data as literal characters. For example, the modeller can enter a format for part-no as @@@-@@@@, to force a part number to appears as LCA-6198. When the user enters in the part number, the hyphens would appear automatically.

- Decimal places definitions

  Definition of decimal places can be used to determine the number of decimal places that should be used to display numbers.

- Boundary definitions (validation rules)

  Validation rules help to ensure that the modelleler is entering valid data. It can be used to limit numeric fields to a certain range. As an example, Conveyor speed could be limited to values no less than 3 ft/sec and not greater than 6 ft/sec.

- Input mask definitions, etc.

  Input mask properties define input masks which aid data entry.

- Not-Null property

Not-null property is used to ensure that data entry in a field is required. As an example, if the modeller tries to add a record without entering the required data field (by mistake), the data entry form view displays an error message.

According to the above discussion, it ispossible to partially automate the verification and validation of input data via well known database validation rules.

5.5 SUMMARY

The proposed methodology can be successfully applied through a simulation life cycle to generate the required entity model, which can then be used to identify and collect valid input data. It provides a tool kit to accelerate the data modelling exercise. The developed activity models can be given significant contributions to stages such as system investigation, problem formulation, objective definition and the level of detail definition. Accordingly, this reason aids the model builder in identifying appropriate data for a simulation model from the initial stages of the project.

This methodology provides a systematic approach to creating a database which can be used to collect required data. The collated data is conveniently stored in a database which can be directly linked into simulation models. This approach can also support other stages of a simulation project. For instance, the need for detailed model documentation can be reduced via the use of standard modules from the IDEF0 model library. Moreover, It is possible to partially automate the validation and verification of input data through boundary and input mask definitions. Recent advances in simulation software, particularly integration via VBA (Visual Basic Application), means in certain situations it is possible to create the entire simulation model automatically.

CHAPTER 6

DATA COLLECTION METHODOLOGY

6.1 Introduction

Simulation models need much data from various information sources. The necessary data, both descriptive and quantitative may in practice be gathered through various methods. For example, through interviews, accessing existing company information sources, direct observations, etc. Usually, large amounts of data needs to be collected through these methods. This chapter focuses on some useful methods of data collection and the benefits of having a MMOD approach to support these methodologies. Finally, this chapter describes some useful methods of data rationalisation techniques. These enable the acceleration of the data collection exercise.

6.2 Methods of data collection

The collection of data for a simulation model involves any combination of the following potential sources of information and knowledge.

- Reading existing documents/ data sources
- Interviewing one or more experts
- Observing the existing system operations

6.2.1 Reading an existing document or data source

A simulation project is only as good as its data preparation. A simulation model may need data from numerous data sources. These data sources can vary from simple manual systems to sophisticated computer based systems.

1. Manual System Data Sources.

(a) People

People are the best source of information because they are more familiar with day to day current system operations. There are some ad-hoc activities in the systems which are not properly documented and are hard to explain, such activities can only be obtained by interviewing experts who are most familiar to the system. Tye and Perera (1997) explained that when a particular workstation is busy, another operator may give assistance, but it is not part of their scheduled job. These kind of ad-hoc activities can often be found in manufacturing systems and experts are the best source of information. The interview process is very useful to gather data regarding these activities.

(b) Paper based documents.

There are usually large amounts of information available on the paper based documents. For example,

- Accounts/payroll records
- Hand drawing of the layout
- Maintenance records
- Process flow diagram
- Process sheets

- Quality control records

2. Computer Based Data Sources

Many companies have computer based data sources and the data groups are often stored in various databases. For instance,

- Accounts/Payroll packages
- AutoCAD
- Computer Aided Process Planning Systems (CAPP)
- MRP/MRPII Systems
- Shopfloor Data Collection Systems (SFDC)

These systems often provide the required data for simulation models. However, most model builders find it difficult to identify reliable data sources for certain data items due to multiple data sources for the same data type and indirect existence of data, as explained in the section 3.2.6. These data sources will never provide erroneous data in both manual and computer based system data sources. It is very remarkably difficult to decide what is the most accurate data source for a certain data item. It depends on modellers experience and how much knowledge and confidence he or she has regarding company data sources. Nevertheless, during the research, major effort has been made to match attributes of reference data with existing data sources, as shown in Figure 6.1. Based on the above result, the conceptual mapping table has been developed to match data sources to developed reference model (See table 6.1). The conceptual mapping table shows how each attribute of the reference data model matches with the company data source according to accuracy and availability of data.

The following methods have been used to rank data sources.

1. Questionnaire survey

2. Talking to manufacturing experts

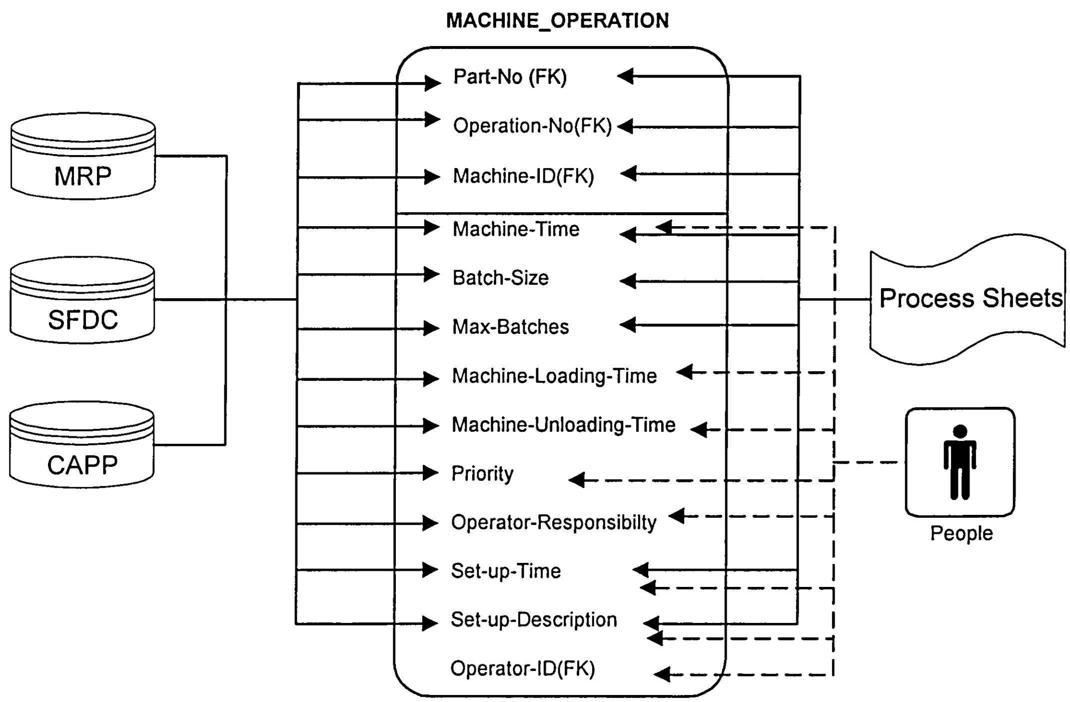


Figure 6.1 Data source matched reference data model

DATA SOURCE	AutoCAD	MRP/ MRP II	CAPP	Shop floor data collection system	Accounts/ Payroll Records	Process Sheets	Process flow Diagrams	Hand drawing of the layout	Maintenance Records	Quality Control Records	People
PART											
Part ID		X	X	X		X	X				
Part Description		X	X	X		X	X				
Required Quantity		1	2	3							
Due Date		1	2	3							
Start Date		1	2	3							
ASSEMBLY_STRUCTURE											
Parent Comp-No		1	2	3							
Child Comp_No		1	2	3							
Assembly Description		1	2	3							
Quantity		1	2	3							
Assembly Level		1	2	3							
MACHINE											
Machine ID	X	X	X	X		X	X	X			
Machine Description	X	X	X	X		X	X	X			
MACHINE_GROUP											
Machine Group Name	X	X	X	X		X	X	X			
Number of Machine	X	X	X	X		X	X	X			
Capacity	X	X	X	X		X	X	X			
MTBF		3	X	1					2		X
MTTR		3	X	1					2		X
Input Buffer Capacity											
Output Buffer Capacity											
MACHINE/OPERATION											
Operation No		X	X	X		X	X				
Manuf Operation Descrp		X	X	X		X	X				
Machine Time		2	3	1		X					X
Scrap Rate		3	X	1		X				2	X
Set-up Time		2	3	1		X					X
Setup description		2	3	1		X					X
Batch Size						X					
Max. Batches						X					
Machine Loading Time		X	X	X							X
Machine Unloading Time		X	X	X							X

Table 6.1 Data Source Match Reference Data Model (1=highest accuracy, 3=least accuracy and X=availability)

DATA SOURCE	AutoCAD	MRP/ MRP II	CAPP	Shop floor data collection system	Accounts/ Payroll Records	Process Sheets	Process flow Diagrams	Hand drawing of the layout	Maintenance Records	Quality Control Records	People
OPERATOR											
Operator ID											
Description											
Efficiency			X	1	X						X
Skills			X	X	X						X
Learning Curve Factor			X	X	X						X
SCHEDULE											
Schedule Name		2	1	3							X
Schedule Description		2	1	3							X
Schedule Duration		2	1	3							X
Machine Available duration		2	1	3							X
Machine Non-available		2	1	3							X
Operator Available Duration		2	1	3							X
Operator Non available Duration		2	1	3							X

Table 6.1 (Continued) Data Source Match Reference Model (1=highest accuracy, 3=least accuracy and X=availabilty)

6.2.2 The Interview Process

Interviews provide useful information about the system to be modelled. IDEF0 modelling diagrams can add significant value to the interview process. These diagrams are able to find facts regarding current operations and problems or objectives to assist further the modelling process. IDEF0 diagrams including a MMOD approach entirely, can be used to determine what information must be gathered during interviews and also to think what problems involve the current system. By reading IDEF0 diagrams, it is easy to get an initial overview of a system and formulate questions for interviews with experts.

Bobak (1997) suggests that a good data model is like a story and by reading the model, the business structure can be described. Reference databases can be used to collect data while the actual interview process is going on. During the interview process, the relevant form and data field can be retrieved from the database according to the type of answer the client presents. The form will then be completed in a sequential manner during the interview with client. The reference database will guide the modeller through the correct steps to collect the required data during the interview.

6.2.3 Observing the existing system operations

Through observation and participation, the modeller can gain a better knowledge of the day-to-day operation of the system to be modelled. During these observations, the modeller can use a pre-built IDEF0 activity modules library to illustrate the activities of the actual system. These diagrams will serve as a reference for the modeller whenever he or she needs to review the facility again.

6.3 Data rationalisation

Data rationalisation can be defined as "any reduction of volume of input data needed by a simulation model". This is particularly true when a large number of products are involved. For example, a typical manufacturing system can have hundreds of different products and thousands of different parts. It is obvious that any reduction of input data needed by a simulation model may accelerate the input data collection exercise.

The aim of this section is to show some useful methods of data rationalisation which accelerates the data collection exercise; it does not provide an exhaustive list. Reducing the volume of input data of a model may lead to a loss in accuracy of output results. Simulation models are therefore used to assess the impact of each discussed data rationalisation technique on output results.

Some guidelines to best practice of methods will be shown in figures 6.2 and 6.3. The intention is to identify interesting portions of the model in which to apply rationalisation and also to determine when these methods can be used.

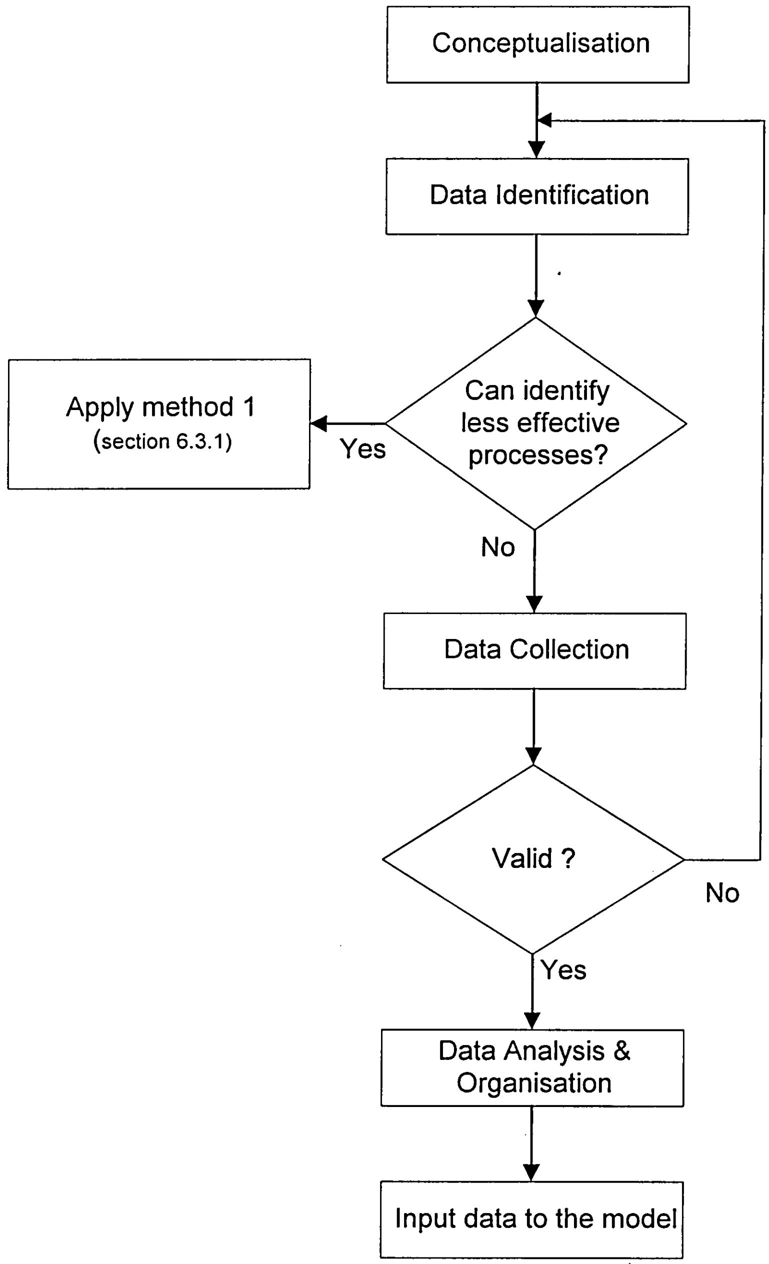


Figure 6.2. A flow chart to explain application of method 1 (Avoid more data collection for less effective process)

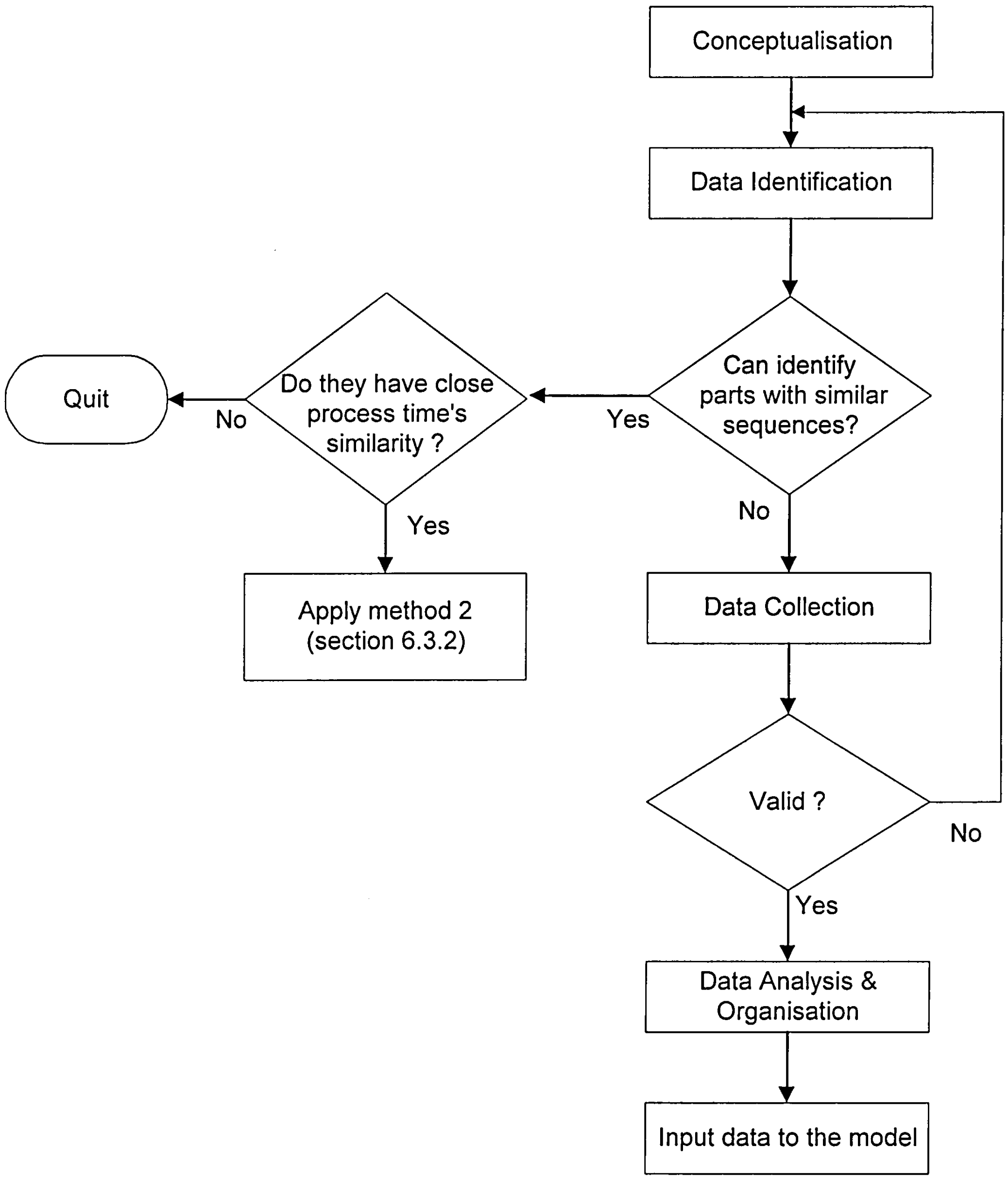


Figure 6.3. A flow chart to explain application of method 2 (Group parts together which have similar process sequences)

6.3.1 Method 1: Avoid more data collection for less effective process

To obtain the best empirical statistical distribution, representing the process time of a resource, the modeller needs to obtain a much larger data sample. However, the time required for the collection of a large data sample is excessively large and this task is often underestimated.

This method describes the avoidance of more data collection for a less effective process and consists of two basic steps.

Step 1 Identify the less effective process in the system which has less impact on the experimental results.

Step 2 Collect less data on these process

Example

The simple hypothetical example, as shown in figure 6.4, consists of three processes and two parts. The parts enter the system one at a time with an exponentially distributed random time between arrivals, with a mean of 20 minutes, and 30 minutes for part 2. The process time for each part on process 1, 2 and 3 are Tria(20,25,30), 2 and Tria(30, 35,40) respectively. Travel time between different machines are assumed to be negligible. The selection rule for each machine is first in first out (FIFO).

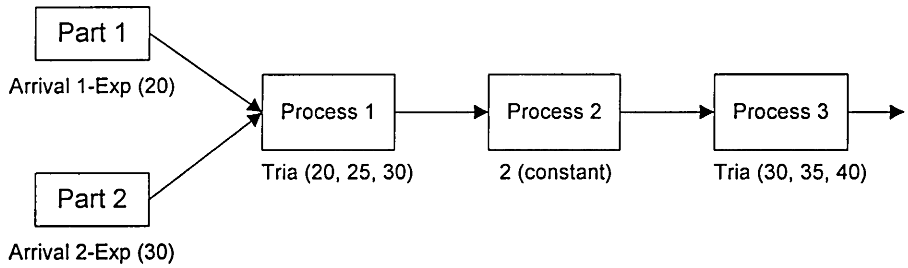


Figure 6.4. Hypothetical example

Sensitivity analysis : The sensitivity analysis for this method can be done using extremes values to check the effect of the result. The simulation model for the above simple example was built using ARENA.

In this case, the used extreme values for the process 2 are 0 and 5. The model run was for 20000 minutes, with three replications to see the effect on key results such as throughput time, resource utilisation and individual queues. Having performed the sensitivity analysis, table 6.2 summarises the output results.

Factor	If Process time = 0	If Process time = 2	If Process time = 5
Avg.throughput Time			
• Part 1	4592.2	4638.6	4616.5
• Part 2	4800.5	4841.7	4813.6
Result on throughput			
• Part 1	273	284	268
• Part 2	150	139	155
Avg. resources utilisation			
• Resource 1	99%	99%	99%
• Resource 2	0	8%	20%
• Resource 3	99%	99%	98%
Avg. queues size			
• Queue at Resource 1	376	394	374.13
• Queue at Resource 2	0	0	0
• Queue at Resource 3	183	185	184.0

Table 6.2 Comparison of output result

Percentage variation of throughput time for Part 1 = 0.47~1.00%

Percentage variation of throughput time for Part 2 = 0.58~.85%

Percentage variation of resource utilisation at process 2 = 0~12%

It can clearly be seen that process 2 will affect the total throughput time of the system only by a small amount. So, it does not matter whether its distribution is exponential, normal and triangular, or constant. It is always easiest to assume as a constant. If throughput time is the required output (performance indicator), then this method can be successfully applied. However, if resource utilisation at process 2 is an important project objective, then this method is not suitable. It is essential to collect more accurate data on process 2 to examine resource utilisation.

6.3.2 Method 2 Grouping parts together

In a medium to high volume batch manufacturing environment, large numbers of product are involved. Rather than modelling each individual part type separately, representative parts can be used to model a group of parts which have a similar process sequence.

This method consists of three basic steps. The first step involves identifying a similar sequence of machine that is visited by each part. Step two is to check any possibility of combining parts in groups according to their close process time similarity. The last step involves a selection of best the statistical distribution to represent a process time for a representative part.

Step 1. Select similar part sequence

Step 2. Group each part according to their processing time's close similarity. i.e. parts which have close process times group together and assign as a representative part. However, this grouping procedure depends on the modeller and the selection of excessively large deviation may lead to the inaccuracy of data that is used in the model. Groups should always, be assigned with accuracy in mind.

Step 3. Replace the processing times of several individual parts with a single processing time for a representative part. The selection of appropriate distributions for process times in representative parts can be done using extremes and mid points of product mix. In most cases, it is best to select triangular distributions for process times. The triangular distribution is suitable because situations which extract the form of the distribution is not unknown but estimates (or guesses) for the minimum, maximum and most likely values are available (Pegden et al, 1990).

Example.

The hypothetical example consists of five different parts and five different machines. Each part type has similar sequences and different process times. The parts enter the system one at a time with an exponentially distributed random time between arrivals and with a mean of 5 minutes. The sample of process times for each part is shown in table 6.3 with their similar process sequences (M1, M2, M3, M4 and M5). All means and standard deviations for each normal distribution have been calculated using random number samples. Travel time between different machines is assumed to be negligible. The selection rule for each machine is first in first out (FIFO).

Part (Qty)	M1	M2	M3	M4	M5
P1 (62)	Norm(23.4,3.09)	Norm(11.2,1.55)	Norm(13.9,2.06)	Norm(18.7,2.58)	Norm(16.4,3.09)
P2 (138)	Norm(41.1,3.61)	Norm(29.2,1.55)	Norm(30.7,2.58)	Norm(19.7,2.58)	Norm(20.4,3.09)
P3 (230)	**Norm(87.8,4.13)**	**Norm(58.9,2.06)**	**Norm(56.4,3.09)**	**Norm(70.4,3.09)**	**Norm(23.9,2.06)**
P4 (320)	Norm(22.4,3.09)	Norm(31.2,1.55)	Norm(28.9,2.06)	Norm(16.7,2.58)	Norm(19.4,3)
P5 (150)	Norm(16.4,3.09)	Norm(26.9,2.06)	Norm(33.4,3.09)	Norm(19.4,3.09)	Norm(18.4,3.09)

Tables 6.3. Process times for example 2

In this example, Part P3 has a larger process time comparatively to others parts. Part P3 is therefore modelled as a separate part and the rest of the parts are grouped as a single representative part with a representative distribution. The simplified model then consists of only two parts. i.e. Part 3 and a Representative part, as described in table 6.4, with their representative process times. The distributions for a representative part were calculated using extremes and mid points of product mix.

Part	M1	M2	M3	M4	M5
Part 3	Norm(87.8,4.13)	Norm(58.9,2.06)	Norm(56.4,3.09)	Norm(70.4,3.09)	Same as Prep part
Prep	Tria(11.34,23,49.3)	Norm(9.30,29,34.7)	Norm(10.5,29,40.4)	Norm(12.4,19,24.4)	Norm(11.3,17,28.6)

Table 6.4. Machine process data for Simplified model

The simulation model for the above example was built using ARENA. In this model, the "CHOOSE" module can be used to route parts to delay required time units, as shown in figure 6.5.

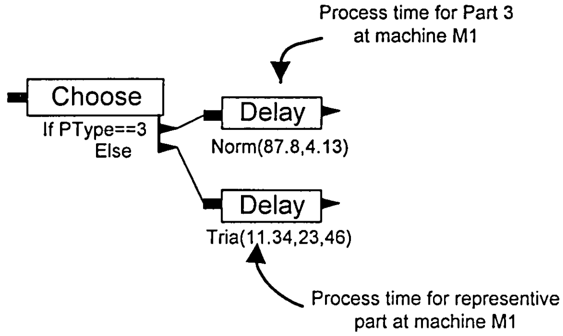


Figure 6.5 Part of a ARENA simulation model

The main experimental results are presented with average values in tables 6.5 to 6.9. All averages are based on three replication runs with a 3000 warm-up period.

System	Part Type	Avg. lead time (min)
Complex	P1	1585.0
	P2	2531.0
	P3	3904.9
	P4	4403.3
	P5	2987.5
Simplify	P3	1159.8
	Prep	3134.7

Table 6.5. Experimental results on avg. lead time

Machine	Complex model(%)	Simplify model (%)	Difference (%)
M1	60.36	62.19	-1.83
M2	84.93	74.29	10.64
M3	85.50	78.36	7.14
M4	75.00	71.79	3.21
M5	99.00	93.00	3.00

Table 6.6 Experimental results on avg. machine utilisation

System	Average WIP in the system
Complex System	375.85
Simplified system	40.13

Table 6.7 Experimental results on avg. work in progress

4. Queues

Queue Name	Average queue size (Complex model)	Average queue size (Simplified model)
M1_ Queue	200.70	275.92
M2_ Queue	86.62	42.91
M3_ Queue	12.522	11.173
M4_ Queue	2.00	13.82
M5_ Queue	66.824	55.02

Table 6.8 Experimental results on avg. work in progress

System	Runtime
Complex system	18518.7
Simplified system	19097.7

Table 6.9 Experimental results on avg. run time

It is obvious that this method is not suitable when the objective involved is lead time. It is more suitable to justify machine utilisation. Guidelines for the use of this method, specifically grouping parts with representative distribution, are not clear cut and depend highly on the modeller.

It can be seen that the investigated data rationalisation techniques may reduce data collection time but selection of techniques is highly dependent on project objectives, as mentioned above.

6.3.3 Advantages and Disadvantages of Methodologies

Advantages

- Reduces the large volume of input data and it makes simulation models more manageable.
- Reduces model complexity and the number of events the model has to handle.
- Reduces data modelling and model building time.
- Improves the run speed of the simulation.
- Provides a way of using less computer memory.

Disadvantages

- Reduces the accuracy of the output data.
- Characteristics such as colour and size can no longer be attributed to individual items.
- The client is much happier if they can see all the parts in the system.

6.4 SUMMARY

Section 6.2 summarised some useful methods of data collection and how the developed MMOD can be used to support fact-finding, problem definitions and pre-interviews, etc. Nevertheless, it may be that in some cases, the data is not available. It is quite common that all the necessary data cannot be obtained from existing data sources. The participants of 1997 Winter Simulation Conference also ranked poor data availability as the most crucial factor. It was considered as a major reason for long data collection time. This may normally happen when modelling a system that does not currently exists. Although, even an existing system can face the same problem as a result of poor data availability, missing data and the impossibility of collecting data. There are various ways of estimating these data, namely, studying similar system, making intelligent guesses, interviewing people who have a good knowledge of a system and discussing the data with equipment vendors.

Some useful data rationalisation methods have been described in section 6.3. It is obvious that these methods can provide less data collection for systems. However, if the modeller has serious doubts about the accuracy and validity of the data and the model, there is no point in using this kind of rationalisation method. These methods can be used to collect less data and rationalise large sets, but care must be taken by the modeller to ensure the accuracy and validation of the data and model.

CHAPTER 7

CONTRIBUTIONS AND FURTHER RESEARCH

7.1 Introduction

Inefficient input data modelling inhibits the rapid development and deployment of simulation models. Most simulation practitioners argue that modelling of input data takes a considerably long time, typically, more than one third of project time. The need for a systematic approach to accelerate the data modelling exercise was identified via an extensive literature survey and an analysis of potential data modelling problems. The main objective of this research is to develop a structured methodology to accelerate the input data modelling exercise within discrete-event simulation of the manufacturing system. The proposed methodology is called MMOD (**M**ethodology for **M**odelling **O**f input **D**ata). An activity module library and a reference data model, both developed using IDEF families of constructs, are the core elements of the methodology. This chapter summarises major contributions made to the field of simulation research and outlines the possibilities for further research.

7.2 Contributions

The lack of research into the process of input data modelling in simulation is surprising given the importance of the study. There are no or little studies that have provided any guidelines on how this process should be carried out. No attempt has been made even to identify the full spectrum of input data modelling problems in the simulation. Consequently, a proper evaluation of problem experiences in input data definition is required. First of all, the study investigated potential problems which influence inefficient data modelling as presented in chapter 3. The research conducted in this thesis has led to address many of these data modelling problems. The modelling problems generally encompass ;

- Wrong or poor problem definitions
- Lack of clear objectives
- Complexity of the system under investigation
- High level of model details
- Poor data availability
- Difficulty in identifying available data sources
- Limited facilities in simulation software to organise and manipulate data

On the basis of a detailed analysis of data modelling problems, the study recommends a methodology to address many of these difficulties. The main contribution to knowledge by this thesis is the development of a **M**ethodology for **M**odelling **O**f **D**ata (MMOD). It provides guidance on the best way of implementation and provides a tool kit to accelerate the modelling exercise. The proposed methodology presented in chapter 4 consists of the following components.

- **Activity module library (IDEF0)**

  The main purpose of activity modules is to show the basic structural behaviour of system resources (machine, people, transporters etc.) and their relationships with entities (parts, material, etc.) from a simulation point of view. The activity model shows the main batch manufacturing functions, which are often modelled in simulation, with their hierarchical structure in terms of the input, outputs, controls and mechanisms. The module library contains generic activity diagrams concerning Machine Operation, Inspection, Material Handling and Storage Operations of batch manufacturing.

- **A reference data model (IDEF1X)**

The objective of the established reference data models is to describe the structure of data that is needed to support batch manufacturing system simulation projects. More formally, it describes how the information is shared by the different activities of batch manufacturing system, which are often modelled in simulation, in terms of data relationships. It provides integration between various input data requirements for components such as part, resources and system logic of a batch manufacturing system, into a single cohesive system to describe a conceptual database implementation.

The resulting reference data model can be converted into a relational database which shows exactly how the entities and relationships will be transformed into an actual database implementation. Most commercial packages available for IDEF modelling can automatically translate an entity model into a relational database. For a relational database, each entity in the reference data model becomes a table and each attribute becomes a column. Primary and foreign keys are declared for each table, and referential integrity constraints are declared for each relationship.

- **Mapping tables**

  The objective of mapping tables is to assist the modeller in identifying the appropriate entities and their attributes in order to generate the required entity models using pre-built IDEF constructs, according to the level of details to be included in the simulation model.

  One of the major shortcomings of the IDEF methodology is the lack of cohesion between activity and data models. This research therefore proposes the development of mapping tables to integrate both the activity and reference data models into a single frame, so that the modeller can identify system activities, corresponding information and data quickly. The proposed mapping tables informally maintain relationships between the activity and data models to overcome the above problem.

The proposed methodology assists the modeller to generate a customised data model (entity model) for the system under investigation. Chapter 5 describes the use of a methodology to generate a customised data model for a given system through the simulation life cycle. The application of the methodology through simulation life cycle also enables the modeller to deal with important phases in the simulation project, such as system investigation, problems and objectives definitions and the level of detail definitions. This approach can also support other stages of a simulation project. For instance, the need for detailed model documentation can be reduced via the use of standard modules from the functional library. Moreover, it is possible to partially automate the validation and verification of input data through boundary and input mask definitions.

The main contribution to knowledge by this thesis is the determination of a comprehensive methodology and guidelines to accelerate the input data modelling exercise in the simulation of discrete-event simulation. This approach offers two major contributions to a simulation project.

- It provides consistency and integrity between the activities and data for a simulation in the early conceptual design stage so that it is easy to decide the amount of input data that needs to be included in a simulation model.

- It provides a flexible design for a customised data model (entity model), according to the knowledge gained from the conceptualisation phase, and the possibility of the translate data model to a fully normalised relational database which can be used to collect the required input data for a simulation.

7.3 Further Research

Currently, the reference model library and data model supports only batch manufacturing environments. Even within these environments, there may be special operational characteristics which have not been incorporated into reference models. It is not practical to develop a methodology to cover eventualities. The best way forward is to develop similar systems for specific manufacturing systems such as aerospace and automobile industries.

Further research efforts could also focus on the automation of the whole procedure. A great deal of work can be done to improve the automation of the developed approach as follows:-

- The main shortcoming of the existing systems modelling methodologies, including the IDEF methodology, is the lack of cohesion between the activity and data models. The proposed methodology in this research informally maintains the relationships between the activity and information models using mapping tables to overcome the above problem. There is need for further research to improve the integration between activity and data models.

- The main extension of this research would be to include the development of interfaces to transform activity modelling diagrams into a simulation model, i.e. automatic extraction of the activities required by a simulation model. The activity modelling diagrams form the basis for identifying data requirements, so that the range of attributes required for a given entity in an entity model is dependent on the level of detail of decomposition, as defined in the activity model. The appropriate data needed by a simulation model can then be collected and stored in user defined databases which are generated from entity models. In an appropriate operating environment, it is possible to link data tables directly into simulation software using recent advanced integration tools. The entire automatic procedure will facilitate in reducing the time necessary to create a validated simulation model. As an example, In the latest version ARENA (version 3.5) can capture process logic of a system which are drawn in the VISIO, i.e. process logic can be transferred from VISIO drawings. The data requirements for the simulation of the process model can be collected and stored in user defined databases which are generated from entity models. The ARENA is capable of integrating databases using reconnect advance integration tools such as VBA (Visual Basic for Applications).

REFERENCES

Ang C L, Luo M, Khoo L P and Gay R K L (1997), A knowledge-based approach to the generation of IDEF0 models, International Journal of Production Research , Vol. 35(5), pp. 1385-1412.

Appleton D S (1995), Business Reengineering with Business Rule/ Ed.:Gover V and Kettinger W J, Business Process Change: Reengineering Concepts, Methods and Technologies, DEA Group Publishing, USA, pp. 291-329.

Ashworth C and Goodland M, (1990) *SSADM,- A practical Approach,* McGraw-Hill, New York., USA.

Au G Paul R J (1997), A graphical discrete simulation environment, Infor, Ottawa, Vol. 3(2), pp.121-137.

Balci O (1990), Guidelines for Successful Simulation Studies, Proceedings of the 1990 Winter Simulation Conference (ed. Balci O, Sadowski R P and Nance R E), New Orleans, USA, pp. 25-32.

Bank J and Gibson R R (1997), Selecting Simulation Software, IEE Solution, May 1997, pp. 30-20.

Baum S S and Glassey P G (1992), Supporting Semiconductor Manufacturing Simulation Tools using a Structured Data Model, Proceedings of the 1992 Winter Simulation Conference (ed.Swain J J, Goldsman D, Crain R C and Wilson), Virginia, USA, pp. 879-884.

Bobak A (1997), Data modelling and design for today's architectures, Artech House, London., pp. 214-228.

Bruce A T(1992), Designing Quality Databases with IDEF1X Information Models, Dorset House Publishing, New York, USA.

Campbell M (1992), Microsoft Access Inside and Out, McGraw-Hill, New York, USA.

Ceric V and Paul R (1992), Diagrammatic Representation of the Conceptual Simulation Model for discrete event systems, Mathematics and Computers in Simulation, Vol. 34, pp. 317-324.

Chen Kai-Ying and Lu Sui-Shong (1996), Integration of Control and Information System for Automated Manufacturing Systems Based on Object-Oriented Approach, Manufacturing Science and Engineering, ASME 1996, pp. 625-632.

Cheng R C H (1993), Selecting input models and random variant generation, Proceedings of the 1993 Winter Simulation Conference (ed. Evans G W, Mollaghasemi M, Russell E C and Biles W E), Georgia, USA, pp. 34-40.

Clark G M (1995), Introduction to Manufacturing Applications, Proceeding of the 1995 Winter Simulation Conference (ed. Alexopoulos C, Kang K, Lilegdon W R and Goldsman), Arlington, USA, pp. 39-45.

Cole J. L. and Valentine P. (1993), Joint Data Base Elements for Modelling and Simulation, Proceedings of the 1993 Winter Simulation Conference (ed. Evans G W, Mollaghasemi M, Russell E C and Biles W), Log Angeles, USA, pp. 971-976.

Date C J (1990), An Introduction to Database Systems, Addison-Wesley Publishing Company Inc, USA.

Date C J (1994), An Introduction to Database Systems-6th edn., Addison-Wesley Publishing Company Inc., USA, Chapter 8.

David B (1997), Information and Data Modelling, 2nd edn, McGraw-Hill., New York, USA.

Denscombe M (1993), An Introduction to Questionnaire Design, Leicester Business School, Leicester, UK

Dietz M (1992), Outline of a Successful Simulation Project, Industrial Engineering, November volume, pp. 50-53.

Douglas B (1997), Entity-relationship modelling and normalization errors, Journal of Database Management, Vol. 8(1), pp.4-12.

Drake G R and Smith J (1996), Simulation Systems for real time planning scheduling and control, Proceedings of the 1996 Winter Simulation Conference (ed. Charnes J.M, Morrice D J, Brunner D T and Swain J J), California, USA, pp. 1083-1090.

Engelike H et al (1985), Integrating Manufacturing Modelling Systems. IBM Journal of Research Development, Vol. 29(4), pp. 343-355.

Evans G W and Golway M W (1994), Simulation of Advanced Manufacturing Systems, Proceedings of the 1994 Winter Simulation Conference (Tew J D, Manivannan, Sadowski D A and Seila A F), Florida, USA, pp. 141-148.

Federal Aviation Administration's (FAA) Office of Information Technology (1995), Business Process Improvement (Reengineering), Handbook of Standards and Guidekines, Version 1.0 (unpublished report), USA

Goodwin A N et al (1989), An assessment of the IDEF notations as description tools, Information Systems, Vol. 14(1), pp. 13-28.

Grabau M R and Maurer R A (1997), Using a Simulation to Generate the Data to Balance an Assembly Line, Proceedings of the 1997 Winter Simulation Conference (ed. Andradottir A, Healy K J, Withers D H and Nelson B L), Atlanta, USA, pp. 733-738.

Grewal S, Tran P and Bhaskare A (1995), ''Assembly Planning Software'', CIRP Annuals Manufacturing Technology, vol. 44(1), pp.1-6.

Hammer M and Champy J (1993), Reengineering the Corporation, Harper Business, New York., USA.

Harrel C R and Field K C (1996), Integrating Process Mapping and Simulation, Proceedings of the 1996 Winter Simulation Conference (ed. Charnes J.M, Morrice D J, Brunner D T and Swain J J), California, USA, pp. 1292-1296.

Harrington H J (1991), Business Process Improvement, McGraw Hill, New York, USA

Hatami S (1990), "Data requirements for analysis of manufacturing systems using computer simulation", Proceeding of the 1990 Winter Simulation Conference (ed. Balci O, Sadowski R P and Nance R E), New Orleans, USA, pp. 632-635.

Hlupic V (1993), Simulation Modelling Software Approaches to Manufacturing Problems, Ph.D. Thesis, University of London, UK

Hlupic V (1998), Business Process Reengineering and Simulation: Bridging the Gap, 12th European Simulation Multiconference, SCS Publication, Manchester, UK, pp 641-645.

Huckvale and Ould Martyn (1995), Process Modelling-Who, What and How: Role activity Diagramming/ Ed.:Gover V and Kettinger W J, Business Process Change: Reengineering Concepts, Methods and Technologies, IDEA Group Publishing, Harrisburg, USA, pp.330-349.

Jayaraman S (1990), Design and Development of an Architecture for Computer-Integrated Manufacturing in the Apparel Industry, Part I: Basic Concepts and Methodology Selection, Textile Research Journal, Vol.60 (5), pp. 247-254.

Jones E (1994), Access 2 Unleashed, SAMS Publishing, Indianapolis, USA.

Joryz H R and Vernadat F B (1990), "CIM_OSA" Part 1- Total Enterprise Modelling and Function view", International Journal of Computer Integrated Manufacturing, vol. 3(3-4), pp. 144-156.

KBSI (1993), '' AI0 Manual, Knowledge Base Systems, Ins, College Station, Texas., USA.

KBSI (1994), '' SMARTER- The Intelligent Information Analysis and Database Design Tool Version 2.0'', User's Manual and Reference Guide, Knowledge Base Systems, Ins, College Station, Texas, USA.

Kelton W D, Sadowski R P and Sadowski D A (1998), Simulation with Arena, McGraw-Hill., New York, USA.

Kettinger W J, Guha S and Teng J T C (1995), The Process Reengineering Life Cycle Methodology: A Case Study/Ed.:Gover V and Kettinger W J, Business Process Change: Reengineering Concepts, Methods and Technologies, IDEA Group Publishing, USA, pp.211-244.

Laamanen M T (1994), The IDEF Standards, Contribution of the US National Institute of Standards and Technology, Methods and Associated Tools for the Information Systems Lifer Cycle, NA-55, USA , pp. 121-130.

Law A M and Kelton W D (1991), Simulation Modelling and Analysis, 2nd edn, McGraw-Hill, New York, USA.

Law A M, McComas M G and Vincent S G (1994), The Crucial Role of Input Modelling in Successful Simulation Studies, Industrial Engineering (July issue), pp. 55-59.

Law A M (1990), "How to build valid credible and appropriately detailed simulation models", Industrial Engineering, April Volume, pp 16-17.

Law A. M. (1993), A Forum on Crucial Issues in the Simulation of Manufacturing Systems, Proceedings of the 1993 Winter Simulation Conference (ed.. Evans G W, Mollaghasemi M, Russell E C and Biles W E),California, USA, pp. 916-922.

Law, A. M., and McComas, M. G.(1997), Simulation of Manufacturing Systems, Proceedings of the 1997 Winter Simulation Conference (ed. Andradottir, A., Healy, K. J., Withers D. H., and Nelson B. L.), Atlanta, USA, pp.86-89.

Leemis L M (1996), Discrete-Event Simulation Input Process Modelling, Proceedings of the 1996 Winter Simulation Conference(ed. Charnes J.M, Morrice D J, Brunner D T and Swain J J), California, USA, pp. 39-46.

Lehtonen J M and Seppala U (1997), A methodology for data gathering and analysis in a logistics simulation project, Integrated Manufacturing System, Vol. 8(6), pp. 65-72.

Lejk M and Deeks D (1998), An Introduction to System Analysis Techniques, Prentice Hall Europe, UK.

Leung W K J and Lai K K (1997), A Structured Methodology to build discrete-event simulation models, Asia-Pacific Journal of Operational Research, May 1997, pp. 121-128.

Lung A.W (1995), ''Develop a Consistent Front-End Approach to Aid Manufacturing Simulation Modelling", 3rd International Conference on Manufacturing Technology, Hong Kong, pp. 455-459.

Lung A W (1998), Benefits of a Systematic approach for Problem Analysis on Manufacturing Simulation Modelling, International Conference on Simulation, University of York, IEE publications, pp. 271-276.

Maji P K (1988), Tool for Development of Information Systems in CIM, Advanced Manufacturing Engineering, Vol. 1, pp. 26-34.

Malhotra R and Jayaraman S (1990), Design and Development of an Architecture for Computer-Integrated Manufacturing in the Apparel Industry, Part II: the Function Model, Textile Research Journal, June Issue, pp. 351-360.

Malhotra R and Jayaraman S (1992), An Integrated Framework for Enterprise Modeling, Journal of Manufacturing Systems, Vol. 11(6), PP. 426-441.

Markowitz H.M (1981), ''Barriers to the Practical Use of Simulation Analysis'', Proceedings of the 1981 Winter Simulation Conference (ed. Oren T I, Delfosse C M and Shub C M), USA, pp 3-9.

Mayer J R, Benjamin P C, Caraway B E and Painter M K (1995), A Framework and a Suite of Methods for Business Process Reengineering/ Ed.:Gover V and Kettinger W J, Business Process Change : Reengineering Concepts, Methods and Technologies, IDEA Group Publishing, USA, pp.245-290.

Mayer R J (1992a), IDEF0 Function Modelling, Knowledge Based Systems, Inc, Texas, USA.

Mayer R J (1992b), IDEF1X Data Modelling, Knowledge Based Systems, Inc., Texas, USA.

Mayer R J and Painter M (1991), IDEF Family of Methods, Technical Report, Knowledge Based Systems, Inc., Texas, USA.

McDonald J E and Donnell J (1995), Joint Database Elements for Modelling and Simulation, Working papers (Electronic Proving Ground, US Army).

Miers D (1994), Use of Tools and Technology within a BPR Initiative/Business Process Re-engineering, ed. Colin Coulson-Thomas, USA.

Moynihan G P (1997), The Application of Enterprise Modelling for Aerospace Manufacturing System Integration, The International Journal of Flexible Manufacturing Systems, Vol. 9,pp. 195-210.

Nordgren, W. B (1995), Steps for proper simulation project management, Proceedings of the 1995 Winter Simulation Conference(ed. Alexopoulos C, Kang K, Lilegdon W R and Goldsman), Arlington, USA, pp. 68-73.

Novels M D (1992), "Modelling the shape of the future", Professional Engineering, June Issue, pp 30-31.

Oakshott Les (1997), Business Modelling and Simulation, Pitman Publishing, London, UK.

Pandya K V (1994), Model for Production Planning and Control Decision at Cell Level: a case study, Computer Integrated Manufacturing Systems, Vol. 7(2), pp 75-92.

Pandya K V, Karlesson A, Sega S and Carrie A (1997), Towards the manufacturing enterprises of the future, International Journal of Operations and Production Management, Vol. 17(5), pp. 502-521.

Pegden C D, Shannon R E, and Sadowski R P (1990), Introduction to Simulation Using SIMAN, System Modelling Corporation, USA.

Peters B A, Smith J S, Curry J and Lajimodiere C, (1996), Advanced Tutorial-Simulation-Based Scheduling and Control, Proceedings of the 1996 Winter Simulation Conference (ed. Charnes J.M, Morrice D J, Brunner D T and Swain J J), California, USA, pp. 194-199.

Pidd M (1992), Computer Simulation in Management Science, 3rd ed., Wiley, Chichester, UK.

Presley A., Huff B., and Liles D H (1993), A Comprehensive Enterprise Model for Small Manufactures, Proceeding of 2nd Industrial Engineering Research Conference, Los Angeles, pp. 664-668.

Ralph R B and Yadav S B (1985), A Methodology to Model the Dynamic Structure of an Organisation, Information Systems, Vol 10(3), pp.299-317.

Ranky, P. G., (1990), Manufacturing database management and knowledge based expert systems, Cimware Limited, England.

Roberts I (1997), BPR for SMEs, Manufacturing Engineer, Vol. 76(6), pp. 261-263.

Robinson S. and Bhatia V.(1995),'' Secrets of Successful Simulation Projects'', Proceeding of the 1995 Winter Simulation Conference (ed. Alexopoulos C, Kang , Lilegdon W R and Goldsman), Arlington, USA, pp. 61-67.

Robinson S., 1994, Successful Simulation, McGraw-Hill International (UK) Limited, UK.

Rohrer M and Banks J (1998), Required Skills of a Simulation Analyst, IEE Solutions (May issue), pp. 20-23.

Sargent R G (1996), Verifying and Validating Simulation Models, Proceedings of the Winter Simulation Conference (ed. Charnes J.M, Morrice D J, Brunner D T and Swain J J), pp, 55-64.

Schormann F and Perera T (1998), Model Representation Methods in Simulation of Manufacturing Systems, International Conference on Simulation, IEE Publication, University of York, pp. 292-296.

Shannon R (1975), Systems Simulation: The Art and Science, Prentice-Hall, USA

Shannon, R E (1998), Introduction to the Art and science of Simulation, Proceedings of the 1998 Winter Simulation Conference (ed. Medeiros D J, Watson E F, Carson J S and Manivannan M S) Washington, USA, pp. 7-14.

Sims, M. J.(1997), An Introduction to Planning and Scheduling with Simulation, Proceedings of the 1997 Winter Simulation Conference (ed. Andradottir, A., Healy, K. J., Withers D. H., and Nelson B. L.), Atlanta, USA, pp. 67-69.

Singh V and Weston R H (1996), "Life cycle support of manufacturing system based on an integration of an tools", International Journal of production Research, 34(1), pp. 1-17.

Srinivasan K and Jayaraman S (1997), Integration of Simulation with Enterprise Models, Proceeding of the 1997 Winter Simulation Conference (ed. Andradottir, A., Healy, K. J., Withers D. H., and Nelson B. L.), Atlanta, pp. 1352-1356.

System Modelling Corporation (1995), Arena User's Guide, Sewickley, USA.

Tatsiopoulos I P (1996), On the unification of bills of materials and routings, Computers inndustry, Vol 32, pp. 293-304.

Trybula W J., (1995), Improving the Modelling Process, Proceedings of the IEEE International Conference of Systems, Man and Cybernetics, Canada, pp. 201-205.

Trybula W.J., (1994), ''Building Simulation Models Without Data'', Proceedings of the IEEE International Conference of Systems, Man and Cybernetics, USA, pp. 209-214.

Tumay K (1995), Business process simulation, Proceedings of the 1995 Winter Simulation Conference (ed. Alexopoulos C, Kang K, Lilegdon W R and Goldsman), Arlington, USA, pp. 55-60.

Tye B. and Perera T., 1997, Simulation Model Design Handbook (unpublished), Sheffield Hallam University, UK.

Vernadat F (1993), CIMOSA: Enterprise Modelling and Enterprise Integration Using a Process Based Approached, Information Infrastructure Systems For Manufacturing, Elsevier Science, pp. 65-79.

Viecent S G and Law A M.,(1993), UniFit II: Total support for simulation input modelling, Proceedings of the 1993 Winter Simulation Conference (ed. Evans G W, Mollaghasemi M, Russell E C and Biles W), California, USA, pp. 199-204.

Vlatka Hlupic (1993), Simulation Modelling Software Approaches to Manufacturing Problems, Ph.D. Thesis, University of London, London, UK

Whitman L, Huff B and Presley A (1997), Structured Models and Dynamic Systems Analysis: The Integration of The IDEF0/IDEF3 Modelling Methods and Discrete Event Simulation, Proceedings of the 1997 Winter Simulation Conference(ed. Andradottir, A, Healy, K. J., Withers D. H., and Nelson B. L), Atalanta, USA, pp. 518-524.

Williams, E. J. and Narayanaswamy, R. (1997), Application of Simulation to Scheduling, Sequencing, and Material Handling, Proceedings of the 1997 Winter Simulation Conference (ed. Andradottir, A., Healy, K. J., Withers D. H., and Nelson B. L.), Atlanta, USA, pp. 861-865.

Wilson, J. R. (1997), Modelling dependencies in stochastic simulation inputs, Proceedings of the 1997 Winter Simulation Conference (ed.. Andradottir, A., Healy, K. J., Withers D. H., and Nelson B. L.), Atlanta, USA, pp. 47- 52.

Wood P J and Johnson P N (1989), "A review of the use of SSADM and IDEF at the University of Warwick", Proceedings of the SAMT'89 Conference, UK.

Wu B (1996), Manufacturing System Design and Analysis, Chapman & Hall, London, UK.

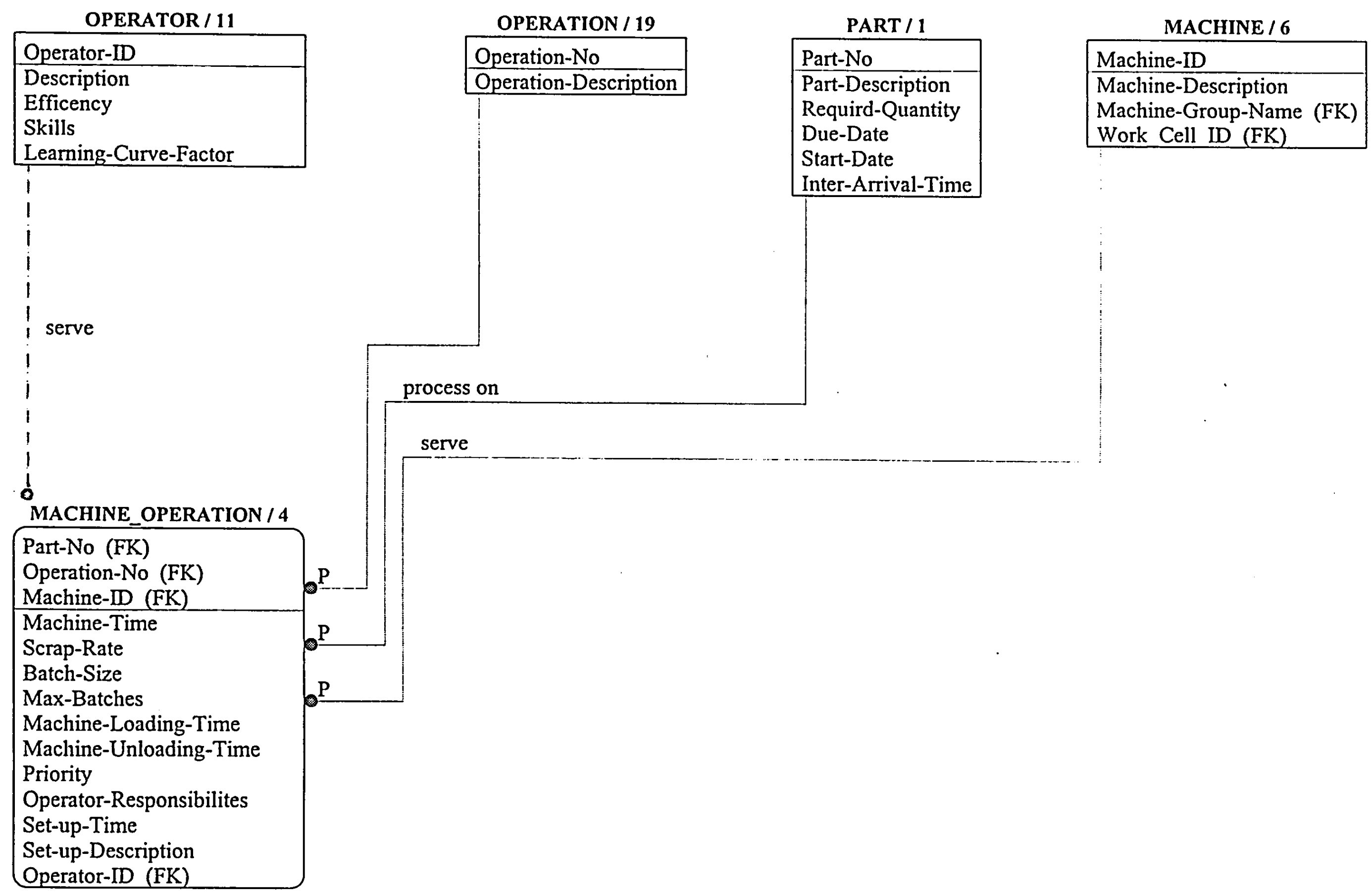
Wyatt T and Al-Malik I (1990), "Methods in manufacturing system engineering-the background", Integrated Manufacturing Systems, Vol. 1(2), 91-92.

Zachman J A (1987), A Framework for Information Systems Architecture, IBM Systems Journal, Vol 26(3), pp. 276-292.

APPENDIX A

Complete views of a reference data model (RDM)

Used At: Thesis
Author: Kapila Liyanage
Project: Reference Data Model
Notes: 1 2 3 4 5 6 7 8 9 10
Date: 3/1/99
Rev: 1
Time: 19:10:06
x Working
Draft
Recommended
Publication
READER
DATE
Context



Node:
Title: RDM [Database]: Machine Operation for non-assembly system (View 1)
Number: Pg 0

Used At: Thesis

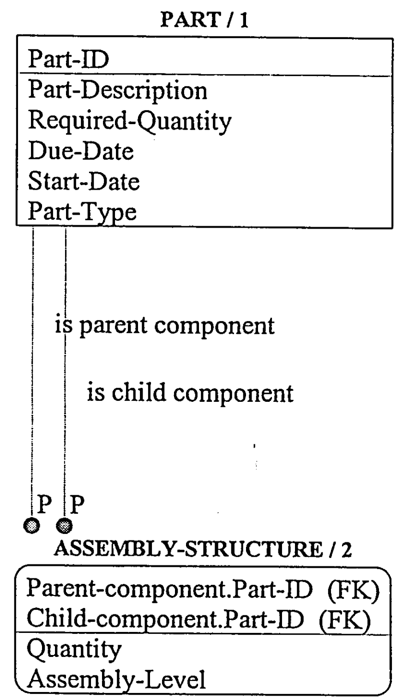
Author: Kapila Liyanage
Project: Reference Data Model
Notes: 1 2 3 4 5 6 7 8 9 10

Date: 3/1/99
Rev: 1
Time: 19:13:07

x Working
Draft
Recommended
Publication

READER | DATE

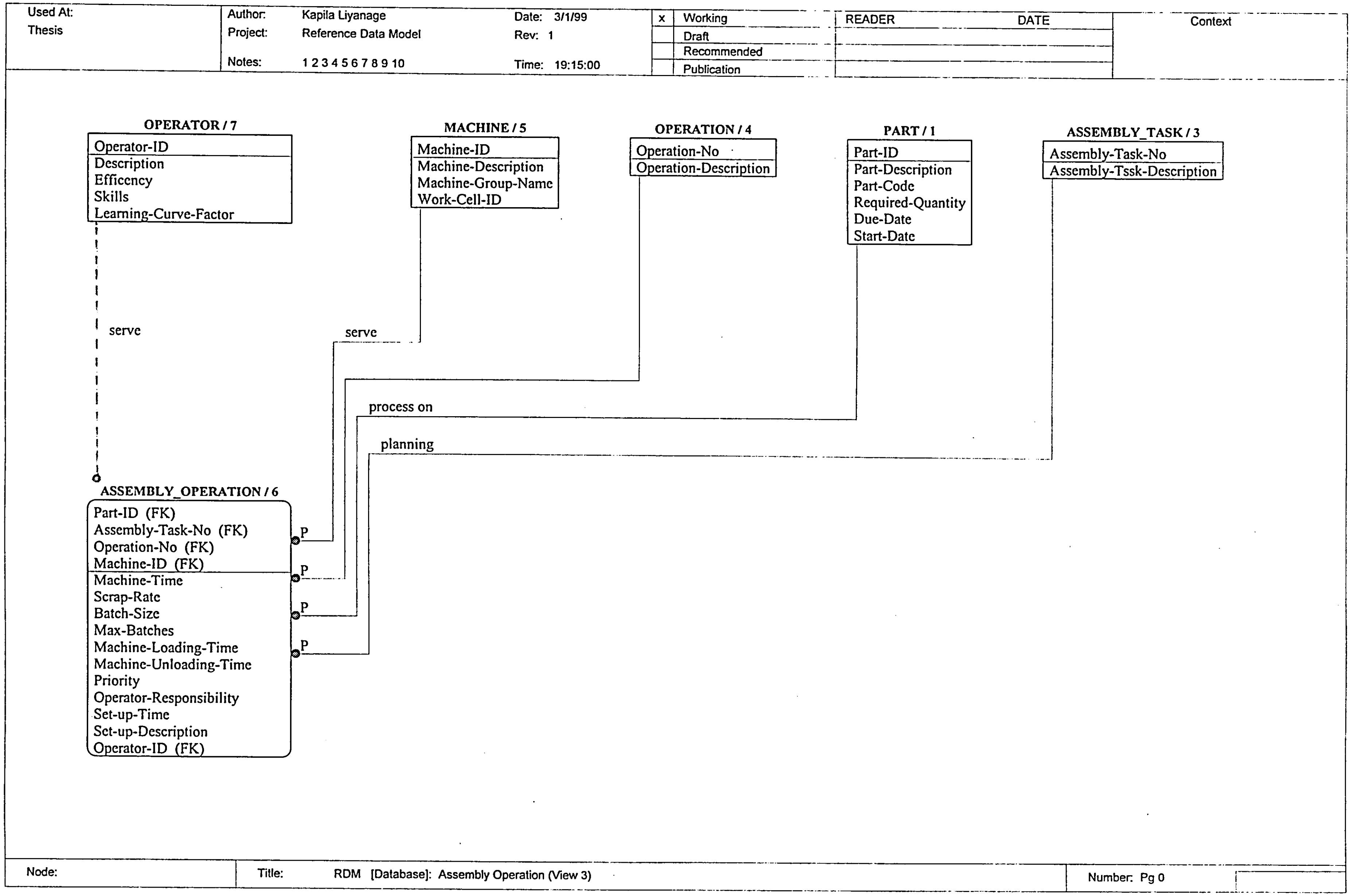
Context



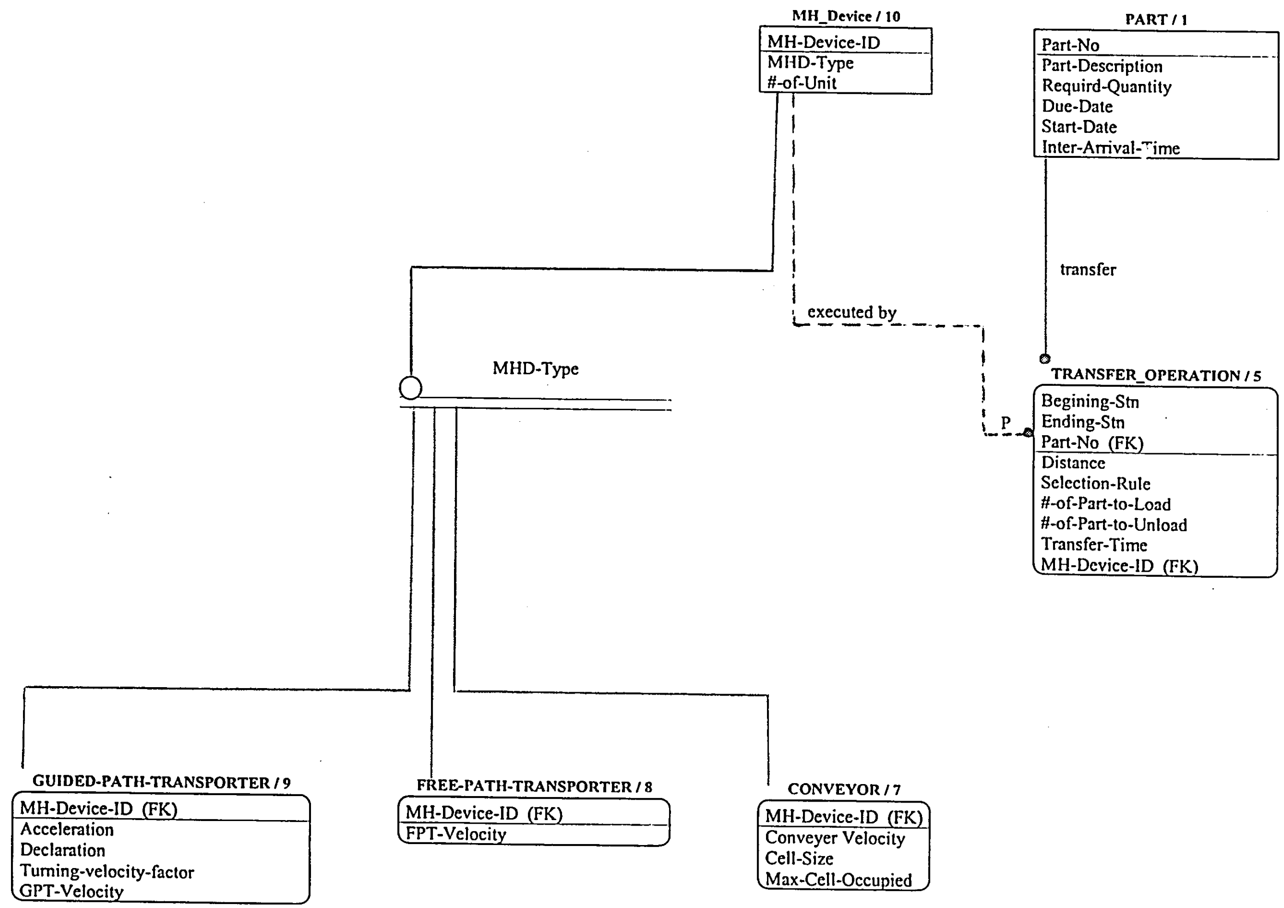
Node:

Title: RDM [Database]: Definition of BOM Structure (View 2)

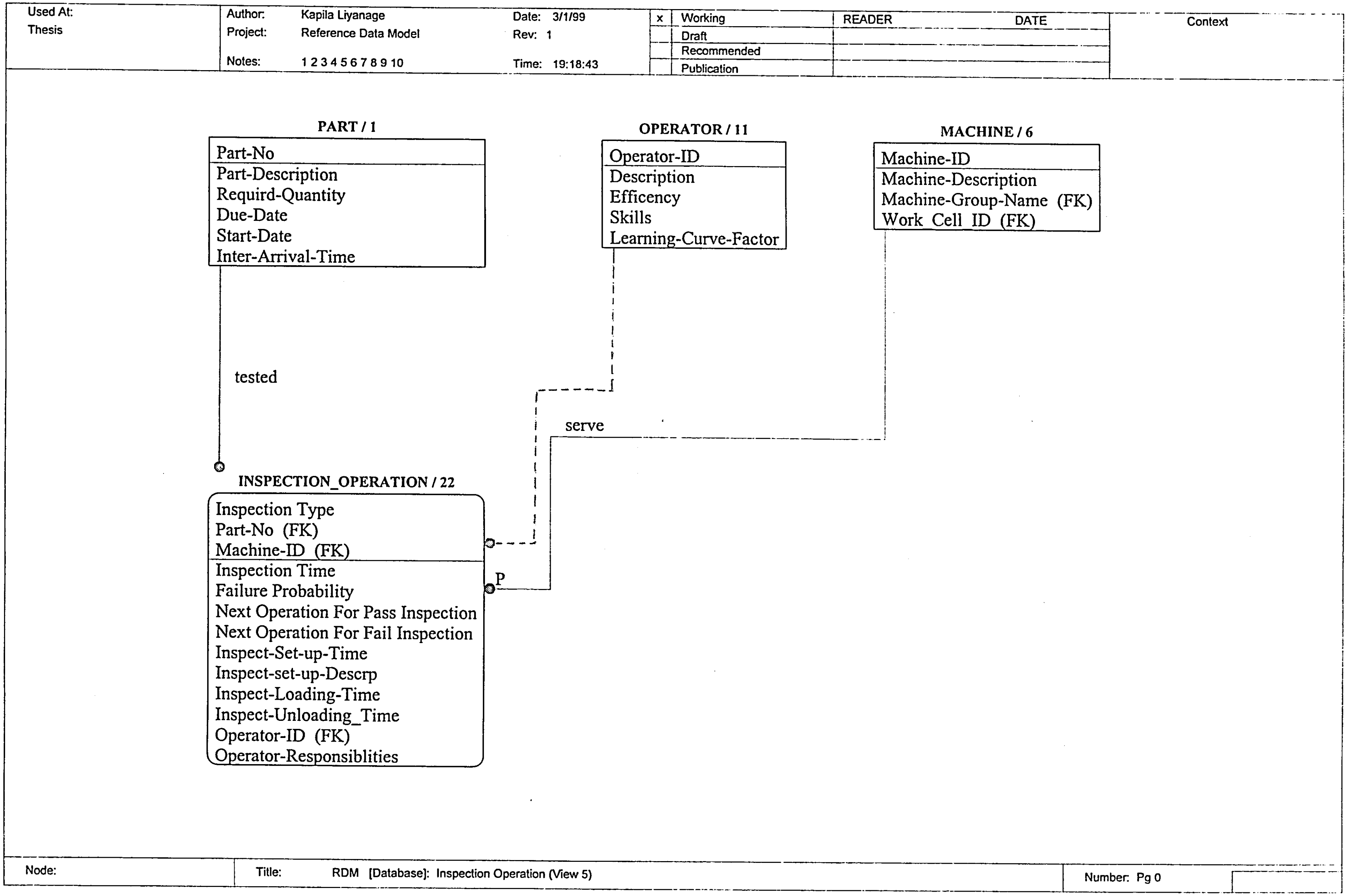
Number: Pg 0


Used At:
Thesis
Author: Kapila Liyanage
Project: Reference Data Model
Notes: 1 2 3 4 5 6 7 8 9 10
Date: 3/1/99
Rev: 1
Time: 19:15:00
x Working
Draft
Recommended
Publication
READER
DATE
Context
OPERATOR / 7
Operator-ID
Description
Efficency
Skills
Learning-Curve-Factor
MACHINE / 5
Machine-ID
Machine-Description
Machine-Group-Name
Work-Cell-ID
OPERATION / 4
Operation-No
Operation-Description
PART / 1
Part-ID
Part-Description
Part-Code
Required-Quantity
Due-Date
Start-Date
ASSEMBLY_TASK / 3
Assembly-Task-No
Assembly-Tssk-Description
serve
serve
process on
planning
ASSEMBLY_OPERATION / 6
Part-ID (FK)
Assembly-Task-No (FK)
Operation-No (FK)
Machine-ID (FK)
Machine-Time
Scrap-Rate
Batch-Size
Max-Batches
Machine-Loading-Time
Machine-Unloading-Time
Priority
Operator-Responsibility
Set-up-Time
Set-up-Description
Operator-ID (FK)
P
P
P
P
Node:
Title: RDM [Database]: Assembly Operation (View 3)
Number: Pg 0

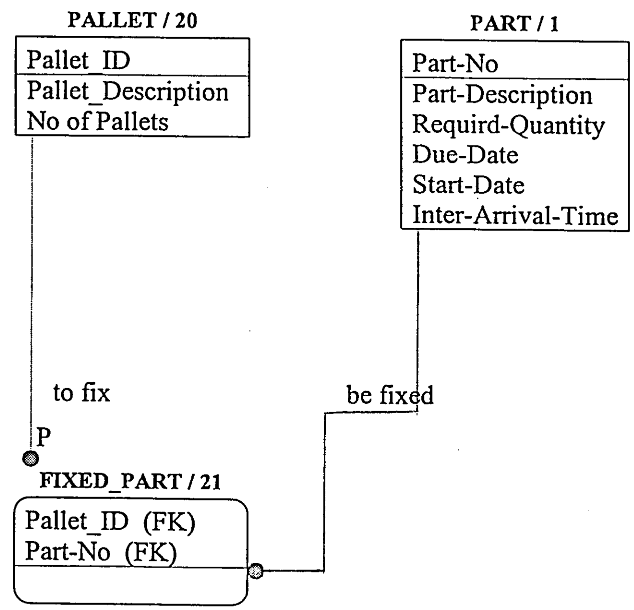
Used At: Thesis	Author: Kapila Liyanage Project: Reference Data Model Notes: 1 2 3 4 5 6 7 8 9 10	Date: 3/1/99 Rev: 1 Time: 19:17:44	x Working Draft Recommended Publication	READER	DATE	Context



Node:	Title: RDM [Database]: Material handling operation (View 4)	Number: Pg 0


Used At:
Thesis
Author: Kapila Liyanage
Project: Reference Data Model
Notes: 1 2 3 4 5 6 7 8 9 10
Date: 3/1/99
Rev: 1
Time: 19:18:43
x Working
Draft
Recommended
Publication
READER
DATE
Context
PART / 1
Part-No
Part-Description
Requird-Quantity
Due-Date
Start-Date
Inter-Arrival-Time
OPERATOR / 11
Operator-ID
Description
Efficency
Skills
Learning-Curve-Factor
MACHINE / 6
Machine-ID
Machine-Description
Machine-Group-Name (FK)
Work Cell ID (FK)
tested
serve
P
INSPECTION_OPERATION / 22
Inspection Type
Part-No (FK)
Machine-ID (FK)
Inspection Time
Failure Probability
Next Operation For Pass Inspection
Next Operation For Fail Inspection
Inspect-Set-up-Time
Inspect-set-up-Descrp
Inspect-Loading-Time
Inspect-Unloading_Time
Operator-ID (FK)
Operator-Responsiblities
Node:
Title: RDM [Database]: Inspection Operation (View 5)
Number: Pg 0

Used At: Thesis	Author:	Kapila Liyanage	Date: 3/1/99	x	Working	READER	DATE	Context
	Project:	Reference Data Model	Rev: 1		Draft			
					Recommended			
	Notes:	1 2 3 4 5 6 7 8 9 10	Time: 19:19:29		Publication			

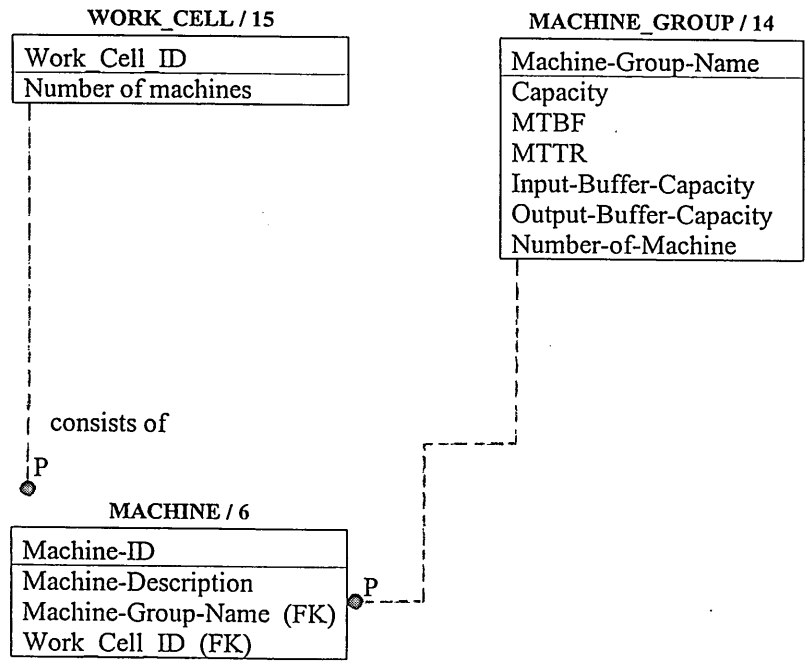


Node:	Title: RDM [Database]: PALLETIZATION (View 6)	Number: Pg 0

Used At: Thesis	Author: Kapila Liyanage Project: Reference Data Model Notes: 1 2 3 4 5 6 7 8 9 10	Date: 3/1/99 Rev: 1 Time: 19:21:56	x Working Draft Recommended Publication

READER DATE

Context

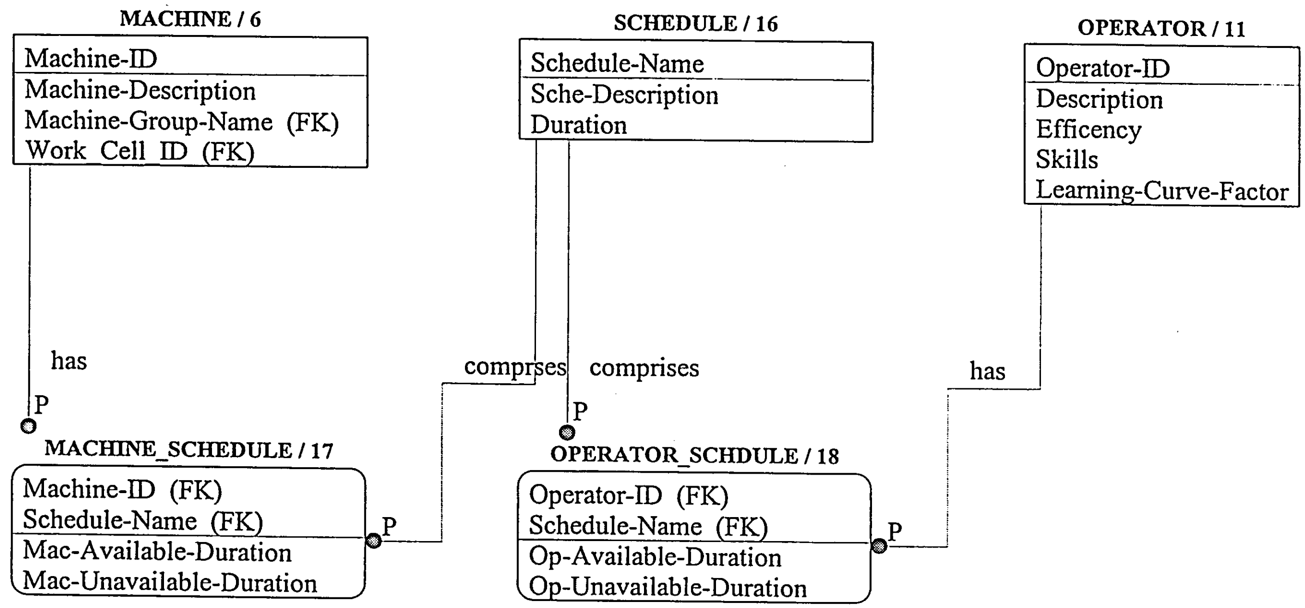


Node:

Title: RDM [Database]: Work cell, machine group and machine (View 7)

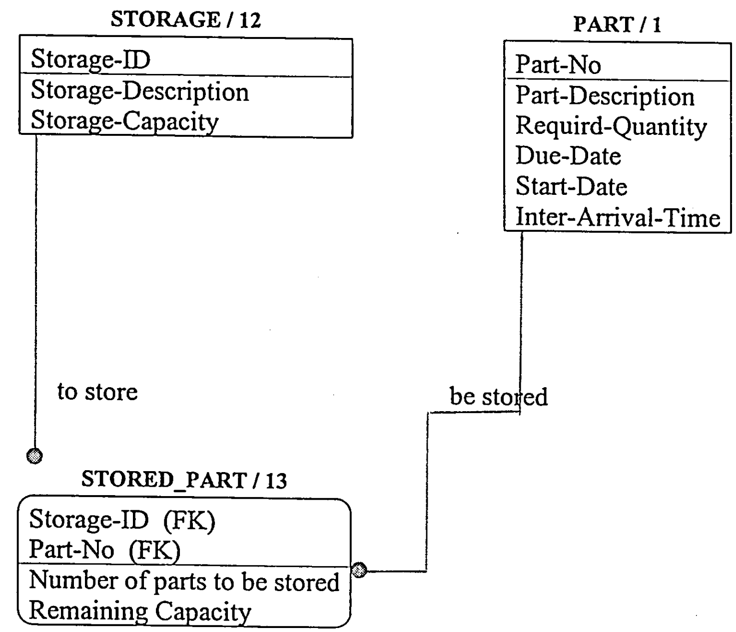
Number: Pg 0

Used At: Thesis	Author: Kapila Liyanage Project: Reference Data Model Notes: 1 2 3 4 5 6 7 8 9 10	Date: 3/1/99 Rev: 1 Time: 19:22:31	x Working Draft Recommended Publication	READER / DATE	Context



Node:	Title: RDM [Database]: Scheduling (View 8)	Number: Pg 0

Used At: Thesis	Author: Kapila Liyanage Project: Reference Data Model Notes: 1 2 3 4 5 6 7 8 9 10	Date: 3/1/99 Rev: 1 Time: 19:23:39	x Working Draft Recommended Publication	READER	DATE	Context



Node:	Title: RDM [Database]: Storage operation (View 9)	Number: Pg 0

APPENDIX B

Questionnaire and analysis of the feedback information the questionnaire

QUESTIONNAIRE- SIMULATION INPUT DATA COLLECTION

(1) What proportion of the total project time is spent on input data collection ?
(Please state approx. value) ....%

(2) Based on your experience, how would you rate the impact of the following factors on the simulation project input data collection time ?

	Very significant						Less significant
Complexity of the system	1	2	3	4	5	6	7
Wrong problem definitions	1	2	3	4	5	6	7
Lack of clear objectives	1	2	3	4	5	6	7
High level of model details	1	2	3	4	5	6	7
Poor data availability	1	2	3	4	5	6	7
Difficult in identifying available data sources	1	2	3	4	5	6	7
Limited facilities in simulation software to organise and manipulate input data	1	2	3	4	5	6	7

(3) Please rank the following items according to the effort required to collect them.

	Less effort required to collect						much effort required to collect
Parts/Materials							
Bill of materials data	1	2	3	4	5	6	7
Schedules data	1	2	3	4	5	6	7
Arrivals time	1	2	3	4	5	6	7
Resources							
Machine time	1	2	3	4	5	6	7
Set-up time	1	2	3	4	5	6	7
Breakdown data	1	2	3	4	5	6	7
Labour allocation	1	2	3	4	5	6	7
Efficiency	1	2	3	4	5	6	7
Rules							
Process routes	1	2	3	4	5	6	7
Priority	1	2	3	4	5	6	7

(4) How do you rank the following data sources in order to obtain accurate/ reliable data on the system being simulated ?

The scale is one(1) to three (3), with (1) being highest accuracy.

	MRP/MRP II	CAPP	Shop floor data collection	Others
Part/Materials				
Bill of materials data				
Schedules data				
Resources				
Machine time				
Set-up time				
Scrap rate				
Breakdown data				
Efficiency				
Rules				
Process routes				
Priority				

(5) When you have large set of similar data (For a example, parts with similar process time/ sequence), do you rationalise them ? If so, how ?

....

....

(6) If data is not collectable, how do you generate it ?

....

....

(7) Which method(s) do you use to store simulation input data ?

- a) Databases [ ]
- b) Paper base systems [ ]
- c) Spreadsheets [ ]

Many Thanks
Please leave the Questionnaire on the chair

Analysis of the feedback information received from the questionnaire

The questionnaire survey conducted at the 1997 Winter Simulation Conference. The Conference is the largest annual gathering of simulation professionals in the world, attended by leading academics and practitioners from all disciplines in simulation. A total of twenty one questionnaires were completed and returned out of fifty questions.

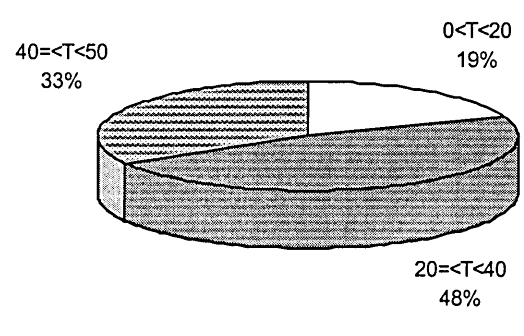
(1) What proportion of the total project time is spent on input data collection ?

(Please state approx. value) ....T....................%

Questionnaire Survey Data for Question 1

Number of Respondents [21]

% Project Time (T)	Number of Respondents	% Value
0< T <20	4	19.05
20-<T<40	10	47.62
40=< T < 50	7	33.33



(2) Based on your experience, how would you rate the impact of the following factors on the simulation project input data collection time ?

Questionnaire Survey Data for Question 2

***Number of Respondents* [20]**

j(=1,2,..7)	i (=1,2,....7)	1	2	3	4	5	6	7	Total No of responded for each factor (TnoR) $\sum_{i=1}^{7}[Xij]$, if j=1,2,..7	I*X= $\sum_{i=1}^{7} i[Xij]$, if j=1,2,..7	Average (I*X/TnoR	Rank
1	Complexity of the system	0	1	2	4	7	1	2	17	79	4.65	**4**
2	Wrong problem definition	5	3	3	3	2	2	0	18	54	3.00	**7**
3	Lack of clear objectives	2	4	3	2	3	1	3	18	69	3.83	**5**
4	High level model details	0	1	2	4	4	3	4	18	90	5.00	**2**
5	Poor data availability	0	0	0	1	3	5	10	19	119	6.26	**1**
6	Difficult in identifying available data sources	0	0	6	1	5	4	2	18	85	4.72	**3**
7	Limited facilities in simulation software to organise and manipulate input data	3	5	2	3	3	0	2	18	60	3.33	**6**

i=1 is less significant and i=7 is very significant X_{ij} *=No of responded for j factor in i th rate.*

(3) Please rank the following items according to the effort required to collect them.

<u>Questionnaire Survey Data for Question 3</u>

Number of Respondents [17]

By using the same calculation procedure which was done in question number 2, the following ranking can be derived.

i(=1,2..,7)	1	2	3	4	5	6	7	Averge	Rank
Parts/Materials data									
Bill of materials data	6	3	0	2	1	1	0	2.38	**3**
Schedules data	3	2	1	4	4	0	0	3.29	**2**
Arrivals time	1	2	5	1	3	3	2	4.18	**1**
Resources data									
Machine time	3	4	3	3	3	0	0	2.75	**5**
Set-up time	0	3	5	4	2	1	0	3.53	**4**
Breakdown data	1	0	0	2	2	8	4	5.59	**1**
Labour allocation	0	1	2	2	6	4	2	4.94	**3**
Efficiency	0	2	2	0	2	5	4	5.20	**2**
Rules data									
Process routes	2	4	3	0	4	2	0	3.40	**1**
Priority	3	5	1	2	3	0	1	3.07	**2**

i=1 is Less effort required to collect data and i=7 is much effort required to collect data

(4) How do you rank the following data sources in order to obtain accurate/ reliable data on the system being simulated ?

<u>Questionnaire Survey Data for Question 4</u>

Number of Respondents [11]

By using the same calculation procedure which was done in question number 2, the following ranking can be derived.

Data Category	Data item	Data Sources	*i*(=3,2, 1)			Average	Rank
			3	**2**	**1**		
Parts/Materials data	BOM data	MRP/MRP II	7	0	0	3.00	**1**
		CAPP	3	2	0	2.60	**2**
		SFDC	0	0	6	1.00	**3**
	Schedule data	MRP/MRP II	1	1	0	2.50	**2**
		CAPP	3	1	0	2.75	**1**
		SFDC	0	3	1	1.75	**3**
Resources data	Machine time	MRP/MRP II	0	2	1	1.67	**2**
		CAPP	0	0	2	1.00	**3**
		SFDC	9	0	2	2.64	**1**
	Set-up time	MRP/MRP II	0	2	1	1.67	**2**
		CAPP	0	0	2	1.00	**3**
		SFDC	6	1	2	2.44	**1**
	Scrap rate	MRP/MRP II	0	1	2	1.33	**3**
		CAPP	0	1	1	1.50	**2**
		SFDC	5	1	1	2.57	**1**
	Breakdown data	MRP/MRP II	0	1	2	1.33	**3**
		CAPP	0	1	1	1.50	**2**
		SFDC	6	0	2	2.50	**1**
	Efficiency	MRP/MRP II	0	2	1	1.67	**2**
		CAPP	1	0	1	2.00	**1**
		SFDC	2	0	4	1.67	**2**
Rules data	Process routes	MRP/MRP II	3	0	1	2.50	**1**
		CAPP	2	0	1	2.33	**2**
		SFDC	0	3	1	1.75	**3**
	Priority	MRP/MRP II	1	0	2	1.67	**3**
		CAPP	0	1	1	1.50	**2**
		SFDC	3	1	1	2.40	**1**

The scale 'i' is three (3) to one (1), with (3) being highest accuracy.

Q5. When you have large set of similar data (For a example, parts with similar process time/ sequence), do you rationalise them ? If so, how ?

Questionnaire Survey Data for Question 5

Total Number of Respondents [10]

No of Respondents	Rationalise large data set	Method(s)
4	yes	• Basis of the process similarity • Self defined method- (group parts routes)
6	No	---

Q6. If data is not collectable, how do you generate it ?

Questionnaire Survey Data for Question 6

Total Number of Respondents [11]

No of Respondents	Method(s)
8	• Using expert knowledge • Intelligent Guess • Statistical modelling

Q7. Which method(s) do you use to store simulation input data ?

Questionnaire Survey Data for Question 7

Total Number of Respondents [18]

Method(s)	No of respondents	% value
Database	9	50
Paper base System	3	16.7
Spreadsheets	16	88.9

1. Liyanage K. and Perera T. (1997),Rapid Data Modelling Techniques in Simulation of Manufacturing Systems, 5th International Conference on FACTORY 2000, IEE, UK.

2. Liyanage K. and Perera T. (1998), Rapid Data Modelling in the Simulation of Manufacturing Systems, ESM'98, 12th European Simulation Multiconference, University of Manchester, UK

3. Liyanage K. and Perera T. (1998), Rapid data collection in the simulation of manufacturing systems, 14th National Conference on Manufacturing Research, University of Derby, UK.

4. Liyanage K. and Perera T. (1998), Design and Development of a Rapid Data Collection Methodology , International Conference on Simulation, University of York, UK.

5. Perera T. and Liyanage K. Methodology for Rapid data Identification and Collection in the Simulation Manufacturing Systems, International Journal of Production Research (Paper submitted and Awaiting referees reports).