CRANFIELD UNIVERSITY

AITOR OYARBIDE-ZUBILLAGA

Manufacturing systems simulation using the principles of System Dynamics

SCHOOL OF INDUSTRIAL AND MANUFACTURING SCIENCE

PHD THESIS

CRANFIELD UNIVERSITY

SCHOOL OF INDUSTRIAL AND MANUFACTURING SCIENCE

DEPARTMENT OF MANUFACTURING SYSTEMS

PHD THESIS

Academic Year 2002-2003

AITOR OYARBIDE-ZUBILLAGA

Manufacturing systems simulation using the principles of System Dynamics

Supervisor: **Dr. Tim Baines**

November 2003

This thesis is submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

Abstract

Manufacturing is the largest single contributor to the global economy. The evolution of consumer demands has pressurised companies into producing a larger variety of products, with improved specifications, reduced costs, and shorter lead times. In this context, companies have found simulation techniques useful in their manufacturing systems design processes; simulation based on Discrete Event Simulation (DES) is the preferred technique. The complexity of manufacturing systems, and the mechanisms of DES, means that the simulation task often consumes excessive time and resources, such as data, software, and training.

Evidence suggests that an alternative modelling technique, named System Dynamics (SD), is also appropriate for conducting this task. SD has been applied successfully in other fields, where its graphical notation is considered beneficial. However, the lack of an SD tool that is tailored toward manufacturing systems has prevented industry from adopting this technique more extensively.

This thesis determines the extent to which SD can provide a credible alternative to DES in the manufacturing system design process. Information concerning DES, SD and practitioners' needs was gathered from published literature and from an interview survey. A functional prototype of a tool based on the SD principles, but tailored to model manufacturing systems was then developed. Three case studies then provided valuable information concerning the requirements of industry and the capabilities of the SD technique.

This research programme has found SD to be sufficiently accurate and quicker than DES tools under certain conditions, requiring less data and skills. In addition, the user interface appears to have had a significant impact on the lack of adoption of SD techniques within the manufacturing sector. Simplifications made by this technique can reduce both model building and model execution time, and thus, experimentation time. However, evidence suggests that DES is still more prevalent, and that further work is required to develop SD based tools tailored to manufacturing systems. Therefore, this thesis provides a much improved understanding of the capabilities of SD as an aid to manufacturing systems design.

Acknowledgements

"To my parents, for their support and love"

A PhD is a large process in which many people become involved in one or another way. As a famous golf player once said: *"only 20% of my actual playing skills are determined by my background; the rest (80%) was achieved by following the right training approach, hard work, and more importantly, by learning from people around me"*. In some way, a PhD does not differ much from this concept, and I would like now to take the opportunity to thank all people who have made this possible.

I wish to begin by thanking the staff of Cranfield University for their help and friendship through the period of this research; always willing to help and making me feel at home.

My most sincere thanks go to my supervisor, Tim Baines. I am extremely grateful, not only for giving me the opportunity to undertake this research, but especially for his support, guidance and friendship. He has always been there; guiding and monitoring the progress, but also motivating me (helped by its inseparable baseball bat) when required.

This is a Basque Government funded project, part of the 'Research Education Programme of the Education Department, Universities and Research'. I would like to thank them for their trust and support.

I am also grateful to all the people involved in this research, especially when conducting the interviews and case studies. Without their valuable inputs, this research would simply not be possible.

A special thank you to colleagues and friends. While the former have shared their experiences with me and making resting time more interesting, the later have dedicate their time and phone bills in order to make my stay in Cranfield more enjoyable and productive.

Last, but not the least, I would like to express deep gratitude to the people to whom this thesis is dedicated: my family. They have not only supported and encouraged me throughout my education, but also sacrificed their lives in order to give me a better one. Thank you very much!

List of Publications

Oyarbide-Zubillaga, A., Baines, T. S., Kay, J. M., and Ladbrook, J. (2000). A system dynamics simulator for manufacture. In: *Proceedings of the 1st World conference on production and operations management*, Seville, Spain, August 27 - September 1 2000, part L1S01. University of Seville, Spain.

Oyarbide-Zubillaga, A., Baines, T. S., Kay, J. M., and Ladbrook, J. (2000). Development of a manufacturing simulator based on system dynamics principles. In: *Proceedings of the 33rd International Symposium on Automotive Technology and Automation: ISATA 2000*, Dublin, Ireland, September 25-29 2000, 95-101. Epsom, UK.

Oyarbide-Zubillaga, A., Baines, T. S., Kay, J. M., and Ladbrook, J. (2002). Manufacturing systems modelling using system dynamics: forming a dedicated modelling tool. In: *Proceedings of the 2nd International Conference on Systems Thinking in Management: ICSTM 2002*, Salford, UK, April 3-5 2002, part A5, 26-32. University of Salford, UK.

Oyarbide-Zubillaga, A. and Baines, T. S. (2003). System dynamics modelling as an aid to manufacturing systems design. In: *Proceedings of the 10th European Concurrent Engineering Conference: ECEC 2003,* Plymouth, UK, April 14-16 2003. University of Plymouth, UK.

Oyarbide-Zubillaga, A., Baines, T.S., Kay, J.M. and Ladbrook, J. (2003). Manufacturing systems modelling using system dynamics: forming a dedicated modelling tool. *Journal of Advanced Manufacturing Systems*, 2(1), June 2003, 71-87.

Oyarbide-Zubillaga, A. and Baines T.S. (2003). System dynamics methodology modelling as an aid to manufacturing systems design; 40 years on. In: *First Joint International Conference: EUROMA-POMS 2003,* Como Lake, Italy, June 16-18 2003. Politecnico di Milano, Italy.

Oyarbide-Zubillaga, A., Baines, T.S. and Kay, J.M. (2003). System dynamics modelling as an aid to manufacturing systems design. In: *17th International Conference on Production Research: ICPR-17,* Blacksburg, Virginia USA, August 3-7 2003. Virginia Polytechnic Institute and State University, Virginia, USA.

Contents

List of Appendices

Appendix A: Systems principles

Appendix B: Stages of a typical simulation project

Appendix C: Interview – Presentation letter

Appendix D: Interview – Questionnaire

Appendix E: Tool's screenshots

Appendix F: Code implemented in the 'design' form

Appendix G: Case study – Presentation letter

List of Figures

List of Tables

Commonly Used Abbreviations

2D / 3D	2 Dimensional / 3 Dimensional
ABC	Activity Based Costing
BP	Business Planning
CAD	Computer Aided Design
CQN	Closed Queuing Network
CT+ / DE+	Continuous Time Plus / Discrete Event Plus
DE+	Discrete Event Plus
DES or DEVS	Discrete Event Simulation
DFD	Data Flow Diagram
DT	Delta Time
GDP	Gross Domestic Product
IDEFx	Icam (Integrated Computer Aided Manufacturing) DEFinition X
IEM	Integrate Enterprise Modelling
IT	Information Technology
MDI / SDI	Multi Document Interface / Single Document Interface
MRP	Material Requirement Planning
O-O	Object oriented
OQN	Open Queuing Network
PN	Petri Net
PSL	Problem Statement Language
QT	Queuing Theory
RP	Rich Picture
SADT	Structure Analysis and Design Technique
SD	System Dynamics
SME	Small and Medium Enterprise
SREM	Software Requirement Engineering Methodology
SSM	Soft System Methodology
ST	System Thinking
VB	Visual Basic

Chapter 1

INTRODUCTION

All industries face pressure to change. There appears to be an ever-increasing market demand for better products, with reduced costs and shorter lead times. This has prompted the continuous search for new and alternative manufacturing system designs. Techniques such as computer based simulation offer a valuable method of testing whether such designs meet expectations. This research is concerned with furthering the use of simulations within the process of manufacturing system design.

This chapter explains the background and context of the research, and summarises why this project has been undertaken. The aim, objectives and the methodology used throughout the project are also introduced. Finally, a brief description of the layout of this thesis is given.

1.1 Overview of research background

The manufacturing system design process follows a number of stages, starting with the identification of a business need, through to the implementation of a new production facility (see Figure 1). A key stage in the design process is evaluation (Mintzberg *et al*, 1976); which usually involves some interaction with both its previous and subsequent stages.

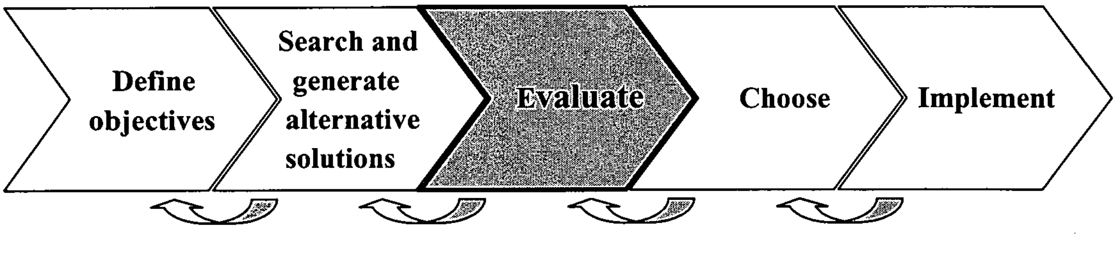


Figure 1: Steps in manufacturing system design (after Baines and Kay, 2002).

Evaluation usually includes an assessment of a manufacturing system, the performance and principal investment. Various analytical tools are available to assist evaluation, such as computer based simulation modelling (Brandimarte and Vila, 1999) (see Figure 2).

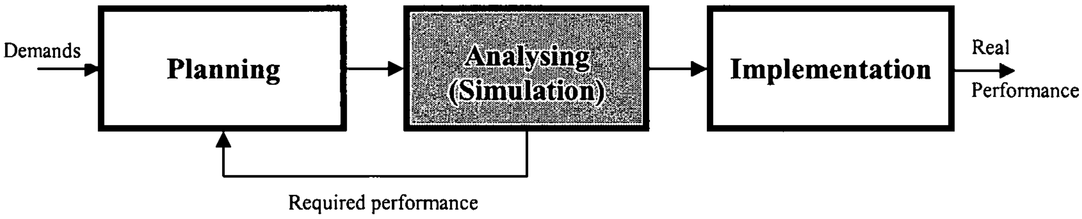


Figure 2: The role of simulation (Brandimarte and Vila, 1999)

Simulation is a modelling technique in which a computer is often used to aid modelling of a real-world system. The computer is programmed to 'mimic' the behaviour of the real system (inputs, outputs and operational logic, see Figure 3), and the model is then used to study and analyse that system. In this way, simulation can be used to answer many types of questions that would otherwise be impossible, risky, or too expensive to answer if tested on the real system.

Simulation is a popular aid used for evaluation within the process of manufacturing system design. Once a model of a manufacturing system has been created, a number of modifications can be made to the model to reflect the different options under consideration. The ensuing model behaviour can then be treated as a prediction of future manufacturing capabilities. There are a number of benefits provided by computer-based simulation that justify the effort invested in this area. The most significant advantage lies in the gain of productivity and throughput time to the process in hand, in combination with a better knowledge of the system prior to its development.

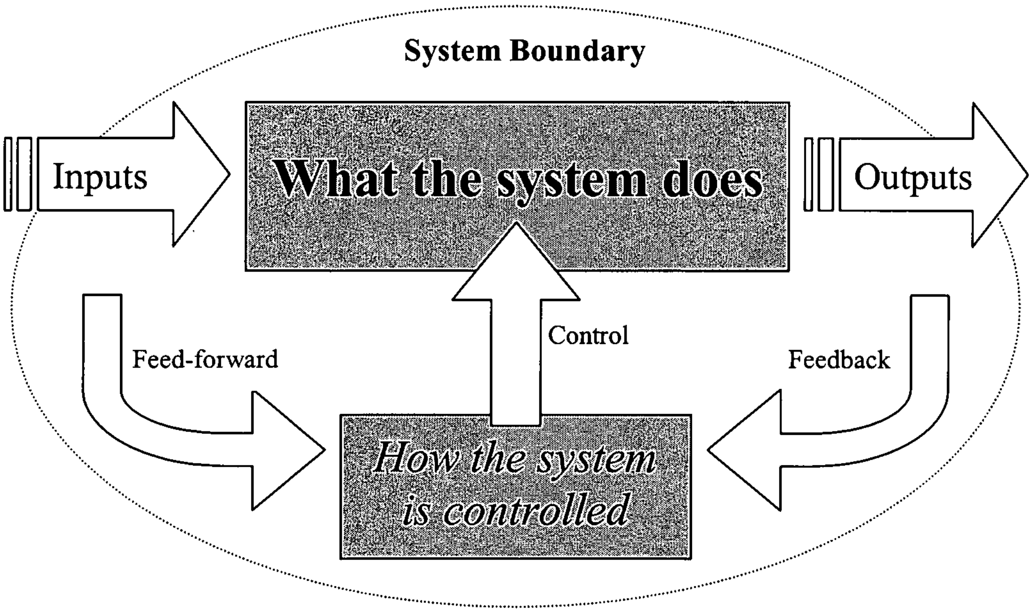


Figure 3: Schematic view of a system and its parts (Terzis, 2001).

Simulation is a symbolic modelling technique (Baines, 1994), and is most often applied in the forms of either Discrete Event Simulation (DES or DEVS) or System Dynamics (SD). DES concerns the modelling of a system as it evolves over time by a representation in which the stated variables change instantaneously at separate points in time (Law and Kelton, 1991). These points in time are when an event occurs, which is an instantaneous occurrence that may change the state of the system. The power and graphical capabilities of Discrete Event Simulator-based tools, such as Witness (Lanner, 2003), Arena (Rockwell Software, 2003) or ProModel (ProModel Solutions, 2003) are widely used for manufacturing systems analysis.

SD is a modelling technique created in the 1960s by Dr. Jay W. Forrester (Ossimitz, 2001) of the Massachusetts Institute of Technology. SD was originally rooted in management and engineering sciences, but has gradually developed into a tool useful in social, economic, physical, chemical, biological, and ecological analysis. The technique is based on treating all interactions within a system as continuous. Whereas DES focuses on activities that start and stop, SD is concerned with states that are changed by flows into and out of the states. These flow rates are themselves controlled by an element termed a rate.

SD can be a useful technique for modelling real world systems. It can be applied by specific tools, such as Stella/iThink (High Performance Systems, 2003), Vensim (Ventana Systems, Inc., 2003), Powersim (Powersim Software, 2003), or Dynamo (Pugh-Roberts Associates, 2003). The analysis of information architectures and the manner in which the information flows have been the primary applications of SD (Ellis, 1998). Unfortunately, there are only a few examples where SD has been applied to manufacturing systems modelling. The 'system' orientation of SD however makes it ideally suited for the analysis of production dynamics, including operational and organisational concept studies.

The SD technique is well suited to modelling complex systems more quickly (Lin, *et al.*, 1997) and has a linear relationship between complexity and computer load requirement (Sterman, 2000). In comparison, DES tends to exhibit an exponential relationship between these factors. When modelling a manufacturing system, which is usually of a complex nature, time is almost always at a premium. It therefore appears strange that DES is almost always the preferred simulation technique used by industrialists. A possible explanation is that DES based modelling tools are initially much more closely aligned with the task of manufacturing system modelling.

Currently available SD based software uses a flow diagram modelling syntax. Although this method of representation has many advantages (Wolstenholme, 1990; Sterman, 2000), it can become very complex and unreadable for the user when simulating manufacturing facilities (Oyarbide *et al*, 2000a). Conversely, although DES based tools require more data, their graphical user interface supports better saleability of the modelling technique. This may well help to explain the greater industrial penetration that these tools have achieved. The research described in this thesis has therefore set out to investigate whether SD can provide a better modelling capability for practitioners if a modelling tool is created that better reflects the challenge of a manufacturing system modelling.

1.2 Overview of research aim and methodology

This thesis has explored whether SD can provide a powerful alternative to DES for manufacturing system design. The research aim is developed in Chapter 3, and is as follows:

> *To determine the extent to which System Dynamics can provide a credible alternative to Discrete Event Simulation in the process of manufacturing system design.*

The main objectives of this research, again as defined in Chapter 3, are outlined below.

1. Understand what is needed of a SD modelling tool for it to suit the needs of a manufacturing system designer.
2. Represent the capabilities of SD in a modelling tool tailored to manufacturing system design.
3. Assess the true capabilities of SD by applying the modelling tool to real manufacturing problems, and assessing performance against a typical DES modelling capability.

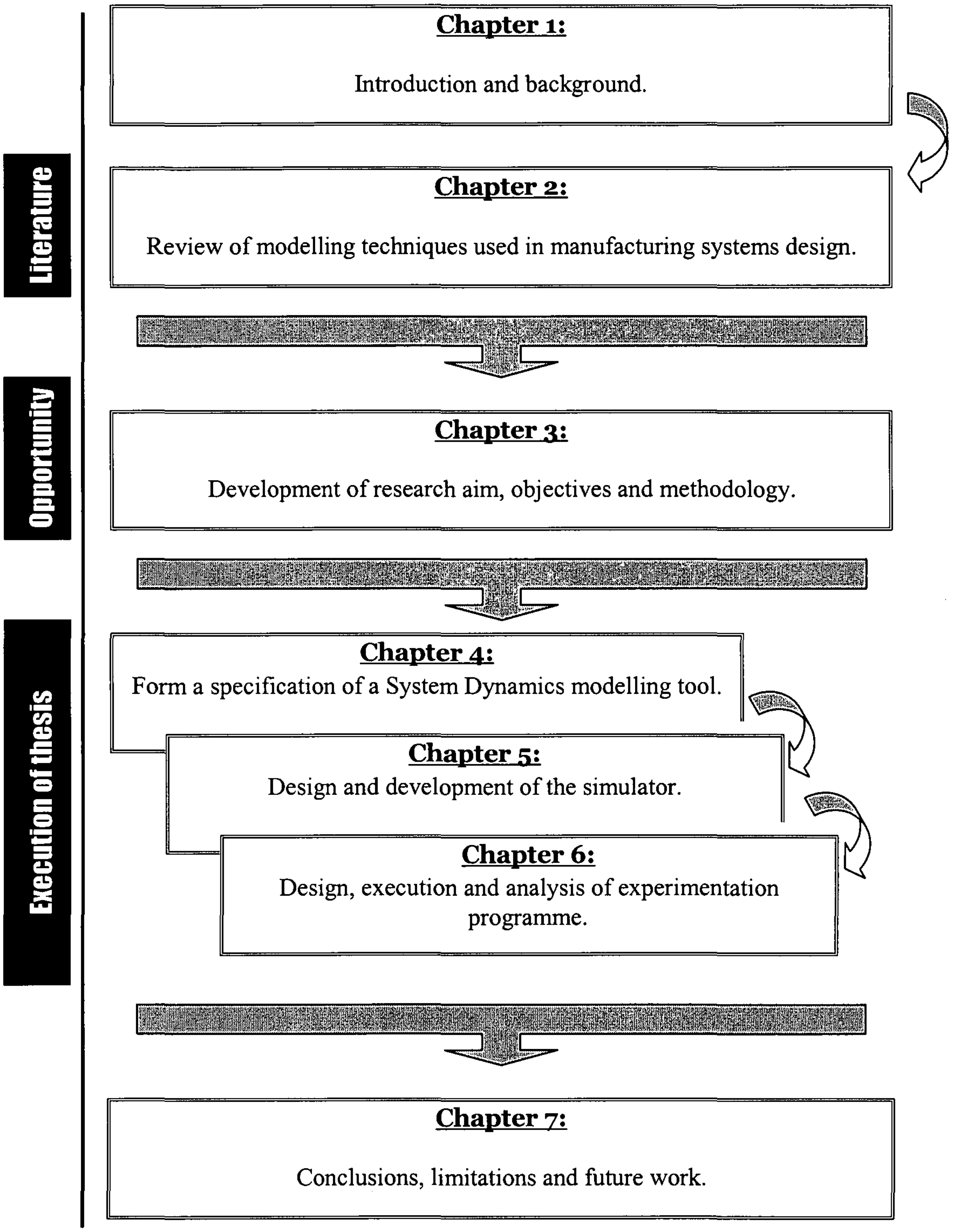
The methodology followed in this project has been based around incorporating the principles of SD into a computer tool tailored to manufacturing system analysis, and then comparing performances against a contemporary DES tool in a number of scenarios. This has enabled the accuracy and utility (e.g. building time, required skills, etc.) of both SD and DES to be compared in a set of case studies.

On the basis of this research programme, this thesis provides a clearer understanding of the actual capabilities of SD, and the opportunities it provides for manufacturing system analysis. The conclusions (Chapter 7) present a range of specific findings that have provided the basis for this contribution.

1.3 Thesis layout

A brief description of the layout of this thesis is given below, and is pictorially depicted in Figure 4.

- **Chapter 2** provides a survey of literature in the area of simulation in the manufacturing industry using analytical techniques. It presents a critical analysis of state-of-the-art DES and SD techniques, the common areas of application, and typical measures obtained in the utilisation of these techniques.
- **Chapter 3** develops the research aim, objectives and scope. It also presents in detail the methodology that is adopted for ensuring that the aim and objectives of this research are attained.
- **Chapter 4** presents the execution of the first stage of the research methodology by describing the industry survey undertaken to obtain the requirements of a manufacturing tailored modelling tool based on SD.
- **Chapter 5** then develops a modelling tool based on the principles of SD, including the main specifications determined from the literature and industry survey. It also highlights the most significant characteristics of the tool.
- **Chapter 6** describes the development of a test bed that is capable of comparing quantitatively both the SD based tool and a commercial DES tool under certain predefined conditions. It also presents the execution of the tests conducted and the results obtained, which are then analysed and discussed.
- **Chapter 7** concludes this thesis with a discussion on the generality of this research, contribution of knowledge, and limitations of the research methodology, developed tool and test bed. It finally discusses the future research directions that follow from this research.



Chapter 1:

Introduction and background.

Chapter 2:

Review of modelling techniques used in manufacturing systems design.

Chapter 3:

Development of research aim, objectives and methodology.

Chapter 4:

Form a specification of a System Dynamics modelling tool.

Chapter 5:

Design and development of the simulator.

Chapter 6:

Design, execution and analysis of experimentation programme.

Chapter 7:

Conclusions, limitations and future work.

Figure 4: Thesis layout

Chapter 2

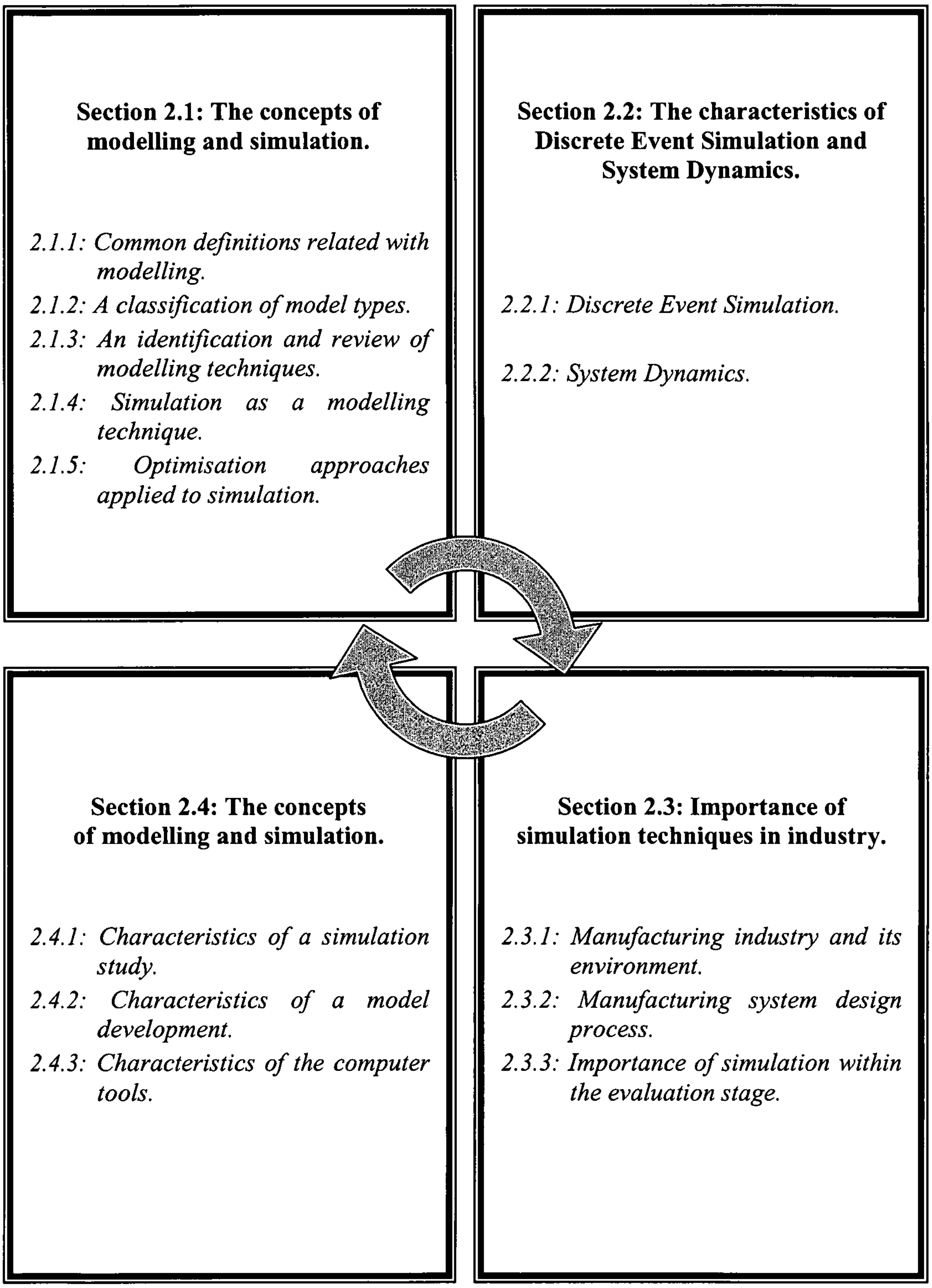
SIMULATION IN MANUFACTURING INDUSTRY: LITERATURE REVIEW

Modelling is a powerful tool; its correct application allows the user to analyse, design and improve the understanding of complex manufacturing systems. In certain cases, modelling can also be used as an efficient communication tool, showing how manufacturing operations work, and helping to stimulate creative thinking about how to improve them. Within modelling, computer simulation has become increasingly popular during recent decades, the increasing power of computers being one of the factors that aid its application within industry.

The intention of this research is to investigate the capability of the simulation technique termed System Dynamics (SD). The purpose of this chapter is to first provide an overview of modelling and simulation, and then to explore SD in more detail, along with the more popular technique of Discrete Event Simulation (DES). This is achieved by addressing the following questions through a review of the literature that has made a valuable contribution to knowledge in this field.

1. What is modelling and simulation, and where does simulation fit within modelling techniques?
2. What are the mechanisms of DES and SD?
3. Why is simulation important and how can it be applied?
4. What are the current research issues that constrain the application of this concept?

The literature review is structured into four sections, as illustrated in Figure 5. First, the concepts and definitions of modelling and simulation are established in Section 2.1, which also provides a categorisation of modelling techniques. Section 2.2 describes the DES and SD simulation techniques and the diagramming conventions associated with them. Section 2.3 highlights the importance of simulation within industry, and addresses how simulation is applied within this context. Finally, Section 2.4 discusses current issues of simulation within manufacturing systems.



Section 2.1: The concepts of modelling and simulation.

- *2.1.1: Common definitions related with modelling.*
- *2.1.2: A classification of model types.*
- *2.1.3: An identification and review of modelling techniques.*
- *2.1.4: Simulation as a modelling technique.*
- *2.1.5: Optimisation approaches applied to simulation.*

Section 2.2: The characteristics of Discrete Event Simulation and System Dynamics.

- *2.2.1: Discrete Event Simulation.*
- *2.2.2: System Dynamics.*

Section 2.3: Importance of simulation techniques in industry.

- *2.3.1: Manufacturing industry and its environment.*
- *2.3.2: Manufacturing system design process.*
- *2.3.3: Importance of simulation within the evaluation stage.*

Section 2.4: The concepts of modelling and simulation.

- *2.4.1: Characteristics of a simulation study.*
- *2.4.2: Characteristics of a model development.*
- *2.4.3: Characteristics of the computer tools.*

Figure 5: Literature review structure

2.1 The concepts of modelling and simulation

This section provides definitions of modelling and simulation, and explores a classification of techniques than can be used when modelling or simulating a manufacturing system.

2.1.1 Common definitions related with modelling and simulation

The word 'system' can be defined in various forms. Due to the variety of ways that it is used, it is difficult to produce a definition sufficiently broad to cover the many uses and, at the same time, sufficiently concise to serve a useful purpose (Gordon, 1978). Similarly, Boucher (1995) says that it is easier to explain what a 'system' is by example rather than a rigorous form of words. Fishman (1973) provides a simple definition of 'system', defining it as *"a collection of related entities, each characterized by attributes that may themselves be related"*. Within manufacturing context, 'systems' may be classified by a number of characteristics (Askin and Standridge, 1993), such as 'open or close', 'adaptive or non-adaptive', 'continuous or discrete', etc.

Depending on the context in which it is used, the word 'model' has also a multitude of meanings. Although the word originally referred to sculpture (for example, a figure), nowadays its meaning is widespread. Merriam-Webster's Unabridged Dictionary (2000) gives seven definitions for the word 'model', for example:

> *... A person or thing that exactly resembles another ...*
>
> *... Usually a three-dimensional representation of something existing in nature or constructed or to be constructed ...*
>
> *... A description, a collection of statistical data, or an analogy used to help visualize often in a simplified way something that cannot be directly observed ...*

Within the manufacturing context, there are almost as many definitions for 'model' as authors (see Table 1). Despite some slightly different interpretations, it seems to be clear that a model is commonly *"some form or abstract representation of a real system that might have a number of uses and be of various forms"*. This thesis adopts this definition, which will now be used in the following sections.

Author	Definition
(Fishman, 1973)	*"A formal representation of theory or a formal account of empirical observation".*
(Gordon, 1978)	*"The body of information about a system gathered for the purpose of studying the system (in the case of physical model, the information is embodied in the properties of the model, in contrast to the symbolic representation in a mathematical model)".*
(Mujtaba, 1994)	... *"A conceptual abstraction of an existing or proposed real system that captures the characteristics of interest of the system".*
(Kheir, 1995)	*"Representation of reality. Reflection of the modeller's understanding of the reality, its components, and their interrelations".*
(Banks *et al.*, 1996)	*"Representation of a system that usually takes the form of a set of assumptions concerning the operation of the system. These assumptions are expressed in mathematical, logical and symbolic relationships between the 'entities', or objects of interest, of the system".*
(Baines *et al.*, 1998)	... *"Emulates the behaviour of a system ... can be used to gain insight into, and make predictions about, that system".*
(Carrie, 1988)	*"A simplified or idealized description of a system, situation, or process, often in mathematical terms, devised to facilitate calculations and predictions".*
(Rubinstein and Melamed, 1998)	*"Abstraction of some real system that can be used to obtain predictions and formulate control strategies".*
(Shannon, 1998)	... *"Representation of a group of objects or ideas in some form other than that of the entity itself".*
(Pidd, 1999)	... *"an external and explicit representation of part of reality as seen by the people who wish to use that model to understand, to change, to manage, and to control that part of reality in some way or other".*
(Banks, 2000a)	*"A model is a representation of an actual system. ... that should be complex enough to answer the questions raised, but not too complex"*
(Sterman, 2000)	... *Models "are 'microworlds', in which decision makers can refresh decision-making skills, conduct experiments, and play".*
(Bellinger, 2002)	*"A simplification of reality intended to promote understanding".*

Table 1: Overview of model definitions (in chronological order)

The term modelling also needs to be clearly understood. The Merriam-Webster's Unabridged Dictionary (2000) defines the word 'modelling' as "... *to plan or form after a pattern ...", or "... to construct or fashion in imitation of a particular model ...*" Manufacturing modelling is a method; it is the process of creating (and often discovering) an understandable simplification of a system that needs to be analysed. A key factor in manufacturing modelling is to maintain the prominent characteristics of the object, concept, or system that it represents (Avni, 1999). When the word 'modelling' appears in this thesis, it refers to the act of creating a 'model'.

Within this context, models exhibit at least one distinctive quality that reflects to the real system (for example, behaviour, geometric dimensions, etc.). Models are usually used to achieve at least one of two major purposes: for explaining and/or understanding the system in hand (descriptive), or to predict and/or duplicate behaviour characteristics (prescriptive) (Shannon, 1975). The latter usually implies the former but not vice versa. Askin and Standridge (1993) expand the number of purposes to five, namely: (i) Optimisation, (ii) Performance prediction, (iii) Control, (iv) Insight and (v) Justification. In either case, any system (even the simpler ones) can be modelled differently depending on the scope and/or level of detail of the study undertaken.

In addition to the definitions of 'system', 'model' and 'modelling', it is important to also explore the terms 'model instance', 'model type, or modelling technique', and 'modelling tool', since these will be used throughout this thesis.

Banerjee and Basu (1993) provide a suitable definition for 'model instance' and 'model type or modelling technique'. They view a model instance as a specific formal representation used in addressing a particular problem, whereas a model type is a possibly infinite collection of model instances characterised by a set of rules and/or properties that distinguish instances of that model type from those of other model types. Hence, when the term 'model' is used in isolation within published literature, and also within this thesis, this is usually an implicit reference to a model instance.

The term 'modelling technique' is used in this thesis when discussing a model type that is directly involved in model construction; therefore, it can be associated with a set of distinguishing properties and rules. Modelling technique can be considered as the principal mechanism that provides the basis for model construction in an operational sense. The variety of modelling techniques that exist and can be applied into the manufacturing context is explained in more detail in Section 2.1.2.

The term 'modelling tool' refers to those tools (usually computer based) that might be required to apply some modelling techniques efficiently in practice. Modelling tools are commonly, but not exclusively, tailored to one specific modelling technique; although it is the underlying technique that is seen to characterise the capabilities of a modelling tool. Table 2 summarises the definitions explained above.

Term	Definition
'System'	Collection of related entities, each characterized by attributes that may themselves be related.
'Model' or 'Model instance'	Some form or abstract representation of a real system that might have a number of uses and be of various forms.
'Modelling'	Process of creating an understandable simplification of a system (model).
'Model type' or 'Modelling technique'	Collection of model instances characterised by a set of rules and/or properties that distinguish instances of that model type from those of other model types. Method used to form/construct a model.
'Modelling tool'	Tool used to apply the concepts of a modelling technique efficiently in practice.

Table 2: Basic definitions related with modelling

2.1.2 A classification of model types

Classification can be defined simplistically and concisely as sorting (Connell, 2000); it can be carried out by addressing concepts such as 'similar features', 'application areas', etc. According to Zopounidis (2002), *"Classification refers to the assignment of alternatives into groups, which are not necessary ordered"*. There are many model types available, and providing a classification of these is valuable.

Unfortunately, literature does not provide a consensus on a form of model type taxonomy. For example, models can be classified simplistically as stochastic or deterministic depending if they are subject to random effects (Banks *et al.*, 1996). Alternatively, a classification can be based on the form of representation. This method is used by Ackoff and Sasieni (1968), who refer to three categories of models, namely: (i) Iconic, (ii) Analogue and (iii) Symbolic.

Shannon (1975) also views models as either iconic, analogue or symbolic. However, in a different way, Schmidt (1985) classifies the models according to the following factors:

1. The manner in which the model describes the system.
2. The purpose of the model.

3. The description of the time dependent behaviour of the system.
4. Description of the random behaviour of elements of the system.
5. The description of system change as a discrete or continuous phenomenon.

Pidd (1988) views models as being either scale, mathematical or logical. He considers computer simulation to be one form of logical model, along with computer flow charts. At the same time, Carrie (1988) refers to three categories of models (iconic, logical and simulation), and Law and Kelton (1991) consider models to be either physical or mathematical, where physical models can also be referred to as iconic models. They suggest that mathematical models are either analytical solutions or simulations.

Mihram (1972) provides extensive work on classifying model types and increases the precision of the work carried out by previous authors by, for example, subdividing symbolic models into three categories (descriptive, simular and formalisations), or considering simulation in a different sub-classification than mathematical models. Table 3 provides a classification of model types according to Mihram (1972).

Authors including Fishman (1973) and Banks *et al.* (1996) agree with this classification. Banks *et al.* (1996) also view models as being mathematical or physical but include simulation as a particular type of mathematical models. They provide three different lower level classifications of model, namely: 'Static or dynamic'; 'Deterministic or stochastic' and 'Discrete or continuous'.

Baines (1994) concludes that the work of Mihram (1972) has provided a foundation against which the views of more recent authors, and evaluations in terminology semantics, can be contrasted. Furthermore, the strength of the classification given by Mihram (1972) is that it goes some way to incorporating most other classifications. The work of Baines (1994) provides a useful update to this work, providing a complementary set of model type definitions (see Table 4). Therefore, this taxonomy has been adopted in this thesis.

Table 3: Classification of model types according to Mihram (Baines, 1994)

		Physical			**Symbolic**		
		Replication	***Quasi-replica***	***Analogue***	***Descriptive***	***Simular***	***Formalization***
Static	***Deterministic***	Earthen relief map	Road map	Statue of B. Franklin	Ten commandments	Decision logic tables	Ohm's law
	Stochastic	Critical dosage test	Weather map	Die toss for Russian roulette	Weather report	Non-adaptive, random chess playing program	Equilibrium queue length
Dynamic	***Deterministic***	Model train set	Planetarium show	Analog computer circuitry	Constitution of the USA	Critical path algorithms	Lanchester's laws
	Stochastic	Drosophila genetic experiment	CRT display of endurance test	White noise generator	Test on Darwinian evolution	Vehicle-by-vehicle transportation model	Stochastic differential equation

(From top to bottom): Increasing generality.

(From left to right): Increasing abstraction, Increasing inferential facility, Decreasing realism.

Main class	Sub-class	Definition
Physical	***Replication***	A spatial transform of an original physical object in which the dimensionality of the modelling is retained in the replica
	Quasi-replica	A physical model in which one or more of the dimensions of the physical object are missing or modified
	Analogue	A model which bears no direct resemblance to the modelled phenomena
Symbolic	***Schematic***	A graphical representation of a system using symbols
	Simulation	A model of the behaviour of a system as a whole by defining in detail how various components interact with each other
	Mathematical	Explicit analytical formulae describing known relationships.

Table 4: Taxonomy of model types (Baines, 1994)

2.1.3 An identification and review of modelling techniques

A modelling technique has been defined in Section 2.1.1 as a mechanism that provides the basis for model construction. Section 2.1.1 has also established common definitions related with modelling, and Section 2.1.2 provided a taxonomy of models types. This classification can now be expanded to include popular modelling techniques. As mentioned in Section 2.1.2, a 'technique' is a *"method used to form/construct a model"*, whereas a 'type' is simply a *"collection of model instances characterised by a set of rules"*.

Classification of modelling techniques is often done by categorising them into different model types. This classification is especially suitable in this context because generic modelling techniques usually capture the flavour of a specific model type. It also helps in the selection of a suitable technique to address the solution of a problem because it reduces the number of techniques to consider (there are usually several techniques that can be applied for a chosen model type). Unfortunately, this classification can create confusion in those cases where an existent modelling technique spans a number of categories, or when a combination of different techniques is used to provide a cohesive modelling tool for a particular application. Examples of the above are 'Petri-nets (PN)' which is a simulation technique that provides a graphical representation as a schematic model, or can be combined with queuing networks to pair the efficient analysis of the former with the high expressiveness of the latter (Balbo *et al.*, 1988).

The taxonomy given by Baines (1994) provides a classification of modelling techniques by first identifying model type and then associating popular techniques with each type. A concern is that this may lead to an incomplete or limited consideration of the techniques available. However, no other taxonomies are apparent that deal with techniques more critically. Therefore, the approach followed in this thesis has been to adopt the taxonomy given by Baines (1994), but remaining aware of this concern. Table 5, shows the link between model types and modelling techniques explained above.

Main class	Sub-class	Generic modelling technique	Ab. term used
Physical	***Replication***	Model construction using an identical mechanism to that used in the real system under study.	Replica
		Model construction using any mechanism that provides a spatially identical model to the real system under study.	Non-functional replica
	Quasi-replica	Model construction using any mechanism that provides a fully functional scaled model.	Scale
		Model construction using any mechanism that provides a scaled model that lacks functionality.	Non-functional scale
		Model construction using any mechanism that provides a two dimensional scaled model that lacks functionality.	2D non-functional scale
	Analogue	Modelling using an analog computer.	Analog
Symbolic	***Schematic***	Rich Picture	RP
		Integrated Enterprise Modelling	IEM
		Icam DEFinition zero	IDEF$_0$
	Simulation	Discrete Event Simulation	DES
		System Dynamics	SD
	Mathematical	Queuing Theory	QT
		Activity Based Costing	ABC
		Business Planning	BP

Table 5: Generic modelling techniques (Baines, 1994)

The following paragraphs expand Table 5 by briefly describing the modelling techniques discussed above and clarifying their main characteristics. It is not the intention of this thesis to provide a full explanation for each modelling technique, because it can be considered as 'common knowledge', and it is adequately covered within published literature.

2.1.3.1 Physical replication models

Physical replication models have been defined as identical or spatial transforms of a real system. These models usually exhibit a close resemblance with reality, and hence, requiring a low level of abstraction. This implies a limited opportunity for various forms of model to exist. The purpose of these models is to select the type of physical replication that is used in practice. For example, in those cases where functionality is a requirement, aesthetics is combined with functionality to provide a model that is a complete replica of the reality. This allows the user to test the model in similar conditions to the final system. An example of a functional model can be a prototype of a product, such as a camera, that is developed in order to test its capabilities before the product is entered into production. These models usually involve various modelling techniques to achieve the functionality required, usually similar to the one used for the development of the real system under study.

Alternatively, a model can be constructed that lacks functionality, being only necessary to build the visual aesthetics. This form of modelling is common in early stages of conceptualisation of new cars, where 'non-functional replica' models are created to choose the most appropriate styling for the body. In this example, the techniques required to build the model of the car varies substantially from the ones required to build the real car. While thousands of components and operations are required to construct a real car, an aesthetic non-functional replica of it can be developed more quickly by sculpturing and painting, for example, a piece of plastic.

2.1.3.2 Physical quasi-replica models

These models have been defined as physical models with one dimension modified or missing, although they can still be fully functional (O'Reilly *et al.*, 1984). The models allow the study of complex systems through the use of scaled-down system replicas (Young *et al.*, 1984). An example of this are the scale models used in wind tunnels. Models of this category usually allow the user to obtain similar measures than the real system, but using fewer resources. Thus, time and money can be saved not only due to the small size of the model, but also due to the lower requirements of the installations required. Similar to the previous ones, these models can be built with a lack of functionality. In this case, the classification described above distinguishes two different possibilities regarding the dimensions of the model. 'Non-functional scale' models are those models where all dimensions are modelled (examples of this are museum scale models or factory layouts (Carrie, 1998)); whilst '2D non-functional scale' models are those where one dimension has been removed (examples of this are photographs).

2.1.3.3 Physical analogue models

Physical analogue models are those that exhibit a similar functionality to the real system being studied, but are not related physically. This characteristic makes the abstraction of these models higher than the previous ones, due to the lack of direct visual relationship with the real system.

Ackoff and Sasieni (1968) cite an example of an analogue model being a hydraulic system representing electrical, traffic, and economic systems. Within the manufacturing context, a system could be modelled, for example, using electronic components such as a microcontroller, switches and bulbs. The algorithms embedded in the microcontroller could be programmed in such a way that mimics the behaviour of the system under consideration, using the switches and bulbs to emulate the inputs (parts, etc.) and outputs (states of the machines, buffers, etc.), respectively.

2.1.3.4 Symbolic schematic models

Symbolic schematic models have been defined as a symbolic graphical representation. Within this category is possible to include simple drawings until complex multi-layer IDEF$_0$ diagrams. They all share the graphical abstraction of the system, being substantially higher than in physical models. Modelling techniques that fit into this category are, for example, Data Flow Diagrams (DFD) (De Marco, 1978; Gane and Sarson, 1979; Johansson *et al.*, 1993), Input/Output Analysis (Olsmats *et al.*, 1988; Baumol and Wolff, 1994), Rich Picture (RP) (Checkland, 1981; Mason and Willcocks, 1994; Macias, 1995), Integrated Enterprise Modelling (IEM) (Mertins *et al.*, 1992) and IDEF$_0$ (Colquhoun *et al.*, 1989; Huff *et al.*, 1991; Johansson *et al.*, 1993).

Rich Pictures are part of the Checkland Soft System Methodology (SSM) (Checkland, 1981). There are no strict rules for drawing rich pictures. Whether or not this is an advantage for the user depends on each particular user. For example, Checkland (2000) states that *"producing such graphics is very natural for some people, very difficult for others"*. He also mentions that *"users need to develop skill in making rich pictures in ways they are comfortable with, ways which are as natural as possible for them as* *individuals"*. This technique has the potential of promoting the communication between individuals as well as the understanding of a system 'at a glance', but the lack of strict rules, in addition to the 'personalisation' of the diagrams can be, in some cases, a disadvantage if misunderstandings arise about the meaning of the symbols used. An example of a 'rich picture' diagram is shown in Figure 6.

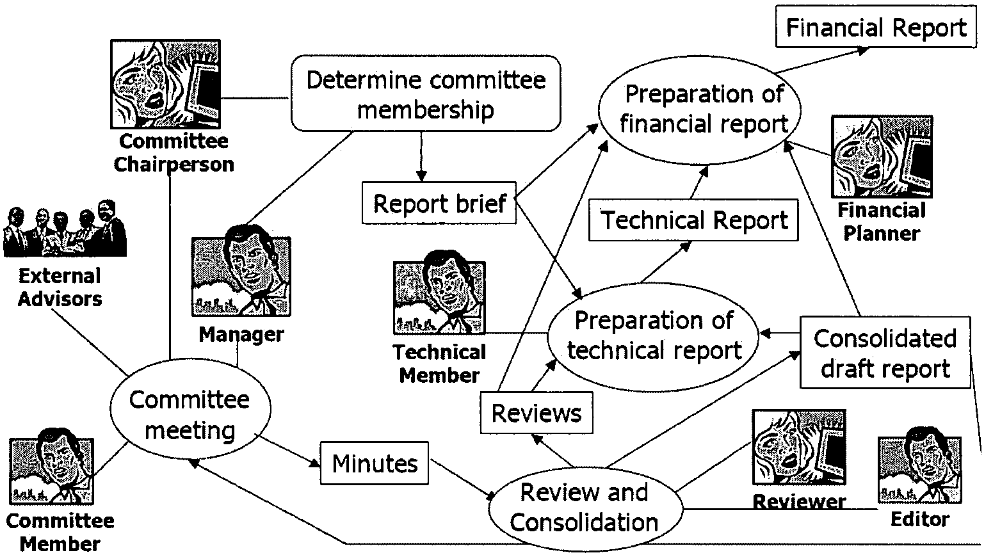


Figure 6: An example of a 'Rich Picture' model (Terzis, 2001).

The Integrated Enterprise Modelling (IEM) method is based on three classes: product, resource and order (Williams, 2000). These classes support the user in the design of the information system architecture and the interfaces in the enterprise (Spur *et al.*, 1996). According to Williams (2000), IEM distinguishes between two views: function model and information model. Tasks on objects and business processes belong to functions and so-called linkage-elements, while the information model describes the data of an enterprise model based on the three classes mentioned above. In addition to the functional and information model, other views can be integrated (for example, control mechanisms, organization units and costs). Although this technique uses stricter rules than RP, these are more flexible than the rules associated with IDEF_O. The strength of IEM models is a consistent representation of complex systems comprising processes, IT-systems and organisation (Edeler and Krause, 1996).

A more popular structured technique within the design of manufacturing systems is the Icam DEFinition zero (IDEF_O), which was developed by the US Air Force under its Integrated Computer Aided Manufacturing (ICAM) programme. IDEF_O is a part of the three methods that were developed to facilitate designing the modelling process (O'Sullivan, 1991). They are IDEF_O (for activity modelling), closely related to Structure

Analysis and Design Technique (SADT); IDEF$_1$ (for data modelling) although this was later expanded to IDEF$_{1x}$ to address databases and IDEF$_2$ (for dynamic modelling of the manufacturing system). At present, seven IDEF standards are used to build the wide range of systems (Rathwell, 2000). They are shown in Table 6.

IDEF	Description
IDEF$_O$	Functional model
IDEF$_1$	Information model
IDEF$_{1x}$	Semantic model (databases)
IDEF$_2$	Dynamic model (simulation)
IDEF$_3$	Process description
IDEF$_4$	Object oriented model
IDEF$_5$	Concept/ontology description
IDEF$_6$	Design rationale model

Table 6: IDEF standards (Rathwell, 2000)

The IDEF$_O$ technique produces a model that is essentially a flow diagram that illustrates the activities within a manufacturing system. It is believed to be typical of a classical approach to structured system analysis and design. Such techniques are characterised by strict rules and considerable abstractions from the system being modelled (Baines, 1994). According to O'Sullivan (1991), *"the principle of IDEF$_O$ is that complex systems can be described in terms of the activities performed in the system and in such a way as progressively to expose detail through a hierarchical decomposition"*. This means that a model begins with an aggregate activities diagram, which is decomposed to expose further detail until the required definition for the system is reached (see Figure 7).

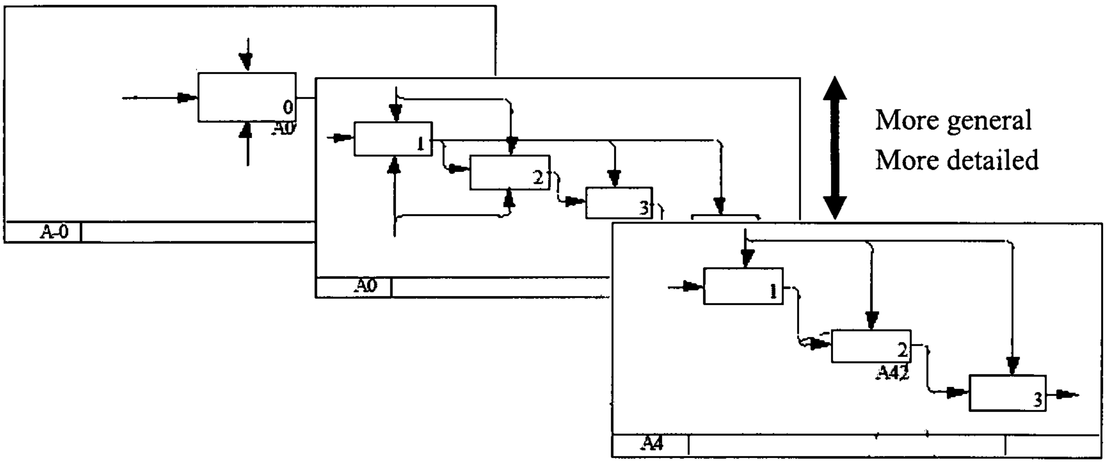


Figure 7: Example of an IDEFo model structure (FED-STD-183, 1993)

2.1.3.5 Symbolic mathematical models

Mathematical modelling has been defined as the use of explicit analytical formulae to describe known relationships. An example of mathematical models is the equation that relates mass, acceleration and force (Olinick, 1978). If a mathematical model can be constructed for a given system, the accuracy and efficiency of mathematical models means that this solution is preferable to alternative modelling methods (Bender, 2000). Mathematical models produce exact answers while simulations only produce approximations, and they are also usually faster. On the contrary, the inflexibility of these models is a disadvantage. Many of the equations involving dynamic processes are formulated as differential equations (Allen, 2002). When an exact solution cannot be obtained, mathematical models can still be used in some situations, by discrete approximation methods such as Euler (Dunham, 1999). In this case, differential equations are translated to difference equations so they can be solved by discrete methods. An example of this concept is as follows (Allen, 2002):

- $dP(t)/dt = k*P(t)$; can be approximated by: $(P_{i+1} - P_i)/\Delta t = k*P_i$; $i = 1, 2, \ldots$

A popular form of mathematical model is Queuing Theory (QT). QT provides a set of mathematical equations that describe the behaviour of a system in order to predict the average behaviour of a manufacturing system over a medium to long time horizon (Suri and Diehl, 1985; Hall, 1990) under specific conditions. Suri and Diehl (1985) state that the overall insight that QT provides is appropriate for the design and planning stage of a manufacturing system. Likewise, Haider *et al.* (1986) state that from their experience, QT models are effective at the initial analysis level of a manufacturing system.

Some mathematical modelling techniques are less frequently associated with modelling. For example, Activity Based Costing (ABC) is a method of measuring the cost and performance of activities and cost objects (Turney, 1996). ABC assigns cost to activities based on their use of resources, and assigns cost to cost objects based on their use of activities. It has been developed to overcome limitations associated with traditional accounting procedures. For example, direct labour is considered another cost pool to be assigned to processes and products in a meaningful manner, not different than any other resource. The primary task of ABC is to assign indirect costs to processes in a manner which better reflects the way in which they are actually incurred (Tarr, 2001). Similarly, Business Planning (BP) models, or 'financial planning systems', are those mathematical modelling techniques that consider the long term performance of a business (Gray, 1984). They are basically a series of projected financial statements about anticipated company financial performance.

2.1.4 Simulation as a modelling technique

Symbolic simulation models are the key topic of this thesis, and so are described in detail in the following sections. To summarise, simulation is concerned with modelling the behaviour of a system as a whole, by defining in detail how various components interact with each other. Merriam-Webster's Unabridged Dictionary (2000) provides several definitions for the word 'simulation' that are also dependent on the context in which the word 'simulation' is used. Examples of these definitions are:

... the act or process of simulating ...

... representation of the operation or features of one process or system through the use of another ...

... the limitation of a physical process or object by a programme that causes a computer to respond mathematically to data and changing conditions as though it were the process or object itself ...

It can be noted that, contrary to the definition of the word 'model', the Merriam-Webster's Unabridged Dictionary (2000) provides some definitions for the word 'simulation' that can be interpreted more easily within a manufacturing context (see definitions 2 and 3). However, none of these definitions seems to be completely accurate when applied to the manufacturing context. Therefore, Table 7 presents an overview of popular key definitions of 'simulation' as provided by experts within this field.

The definitions provided in Table 7 show the lack of consensus when explaining the meaning of the word 'simulation'. This is justified by Robinson (1994) who states that defining simulation is *"in fact a surprisingly difficult question to answer"*. The basic concept that defines 'simulation' as a *"representation of a system by a symbolic model which can be used to execute experiments"* seems to be established and adopted by several authors (Fishman, 1973; Shannon, 1975; Kheir, 1995; Coyle, 1996; Garrido, 2001; etc.).

Author	Definition
(Naylor *et al.*, 1966)	*"Is a numerical technique for conducting experiments on a computer, which involves certain types of mathematical and logical models that describe the behaviour of systems over extended periods of real time".*
(Fishman, 1973)	... *"the act of representing a system by a symbolic model that can be manipulated easily and that produces numerical results."*
(Shannon, 1975)	*"Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies for the operation of the system".*
(Gordon, 1978)	*"Procedure of establishing a model and deriving a solution numerically". "Process of solving the equations of the model, step by step, with increasing values of time".*
(Vemuri, 1978)	*"Simulation is the art of playing around with a simulator".*
(Roberts *et al.*, 1983)	... *"imitation of something". "It generally involves some kind of model or simplified representation".*
(Carrie, 1988)	*"The technique of imitating the behaviour of some situation or system by means of an analogous situation, model or apparatus, either to gain information more conveniently or to train personnel."*
(Law and Kelton, 1991)	... *"numerically exercising a model for the inputs in question to see how they affect the output measures of performance."*
(McHaney, 1991)	*"Simulation is the use of a model to develop conclusions that provide insight on the behaviour of any real world elements". "Computer simulation uses the same concept but requires that the model be created through programming on a computer".*
(Askin and Standridge, 1993)	*"Simulation is the most common method for constructing models that include random behaviour of a large number and a wide variety of components and assess the temporal dynamics of systems"*
(Robinson, 1994)	... *"is a model that mimics reality".*
(Kheir, 1995)	*"Simulation is the process of building and experimenting with (manipulating) a computerised system model such that a specific purpose of the study is achieved through observing the model's behaviour under assumptions defined by the experimenter".*
(Banks *et al.*, 1996)	*"Simulation is the imitation of the operation of a real-world process or system over time"*
(Coyle, 1996)	... *"creation of a set of equations to represent the system and then allow the equations to run forward in simulated time to attempt to mirror the behaviour of the real system"* ... *"in practice, a computer is used".*
(Brandimarte and Vila, 1999)	*"Simulation is the imitation of the real plant in a computer model to deserve the dynamic behaviour under several variants of load and eventual breakdowns"*
(Sterman, 2000)	*"Simulation is the only practical way to test complex models. The complexity of our mental models vastly exceeds our capacity to understand their implications".*
(Garrido, 2001)	... *"larger and more complete model built from the conceptual model, for studying the behaviour of a real system. This model mimics the behaviour of the system, under certain constraints."*

Table 7: Overview of simulation definitions (in chronological order)

However, authors tend to include statements about the techniques and tools used in practice when defining the word 'simulation'. For example, Gordon (1978) includes details about how the equations are solved within the simulation process. Another example can be found in the definition provided by Coyle (1996), who implies the use of a continuous technique when applying simulation. More obvious is the inclusion of the computer tools within the definition of 'simulation' (Naylor *et al.*, 1966; McHaney, 1991; Kheir, 1995; Coyle, 1996). Vemuri (1978) goes further, and states that *"simulation is the art of playing around with a simulator"*. However, exceptions to this exist, and authors such as Fishman (1973) and Shannon (1975) do not explicitly detail the way (methodology or method) in which the simulation is conducted. Thus, this thesis accepts the definition of 'simulation' as *"representation of a system by a symbolic model which can be used to execute experiments, usually aided in practice by a computer"*, which will be used in following sections.

Askin and Standridge (1993) classify manufacturing as discrete parts or continuous processing. Discrete manufacturing is characterized by individual parts that are clearly distinguishable, while process industries operate on product that is continually flowing. Following the taxonomy given by Baines (1994), simulation has been located within the symbolic modelling techniques and has been subdivided into two sub-classes, namely: (i) DES and (ii) SD. In practice, this categorisation is expanded by the inclusion of a 'hybrid' symbolic modelling technique, which combines some features of each of the two techniques described previously (DES and SD) (Barton, 1992; Martin and Raffo, 2001 and Seveance, 2001).

Although analytical and simulation approaches both make use of a model, they 'solve' it in a different manner. As Rubinstein and Melamed (1998) explain, while *"the analytical approach employs strictly mathematical tools to compute various quantities of interest in relatively simple tools"*, *"the simulation approach merely generates possible histories and then calculates statistics from them"*. While this implies the use of statistical estimation (subject to experimental error) by the simulation technique, it also expands the usability and flexibility of the technique to far more complex models.

A reason for the success of simulation techniques within industry is probably the development of powerful computer tools capable of dealing with very complex scenarios in an intuitive manner (Nikoukaran *et al.*, 1998). These computer tools are programmed to 'mimic' the behaviour of the modelled system, and to answer many types of questions that would otherwise be impossible, risky, or too expensive to answer if tested on the real system.

2.1.4.1 Discrete Event Simulation

The DES technique has existed since the 1950s (Robinson, 1994; Radzicki, 1997), and the use of computers to deal with this technique was well established by the 1970s (Fishman, 1973). Discrete Event Simulation (DES) concerns the modelling of a system as it evolves over time by a representation in which the stated variables change instantaneously at separate points in time (Roth, 1987). These points in time are when an event occurs. Here an event is defined as an instantaneous occurrence that may change the state of the system (Law and Kelton, 1991).

The first tool based on Discrete Event Simulation was the 'General Simulation Program (GSP)' written by Tocher in 1958; introducing an alternative paradigm to traditional analytical modelling with random elements (Rubinstein and Melamed, 1998). Mills and Talavage (1985) provide a brief history of the development of these tools and their applicability. Although DES can be applied without a computer, for example by using 'activity cycle diagrams' (Carrie, 1988), nowadays it is mainly applied by using computer based tools or 'simulators'.

2.1.4.2 System Dynamics

Similar to DES, the SD technique has existed since the 1950s (Radzicki, 1997). However, the path followed by DES and SD has been different. Possible reasons for this might be the differences between the techniques themselves, or the manner in which these techniques have been implemented within the computer tools.

There have been numerous definitions of SD (Keys, 1988; Sterman, 2000). The following one is extracted from Wolstenholme (1990):

> *"A rigorous method for qualitative description, exploration and analysis of complex systems in terms of their processes, information, organisational boundaries and strategies; which facilitates quantitative simulation modelling and analysis for the design of system structure and control".*

SD technique is closely related to ideas of Systemic Thinking (ST), being these two terms (SD and ST) sometimes used as synonymous. An example of this can be found in a popular SD based tool, STELLA, which is an acronym for 'Systems Thinking Experimental Learning Laboratory with Animation'. However, this view is not shared by all practitioners. While authors such as Richmond (1994) view SD as *"a subset of* *larger ST framework"*, authors such as Forrester (1994) view ST as *"a door opener for SD"* and even states:

> *"Some people feel they have learned a lot from systems thinking, but they have gone less than 5 percent of the way toward a genuine understanding of systems. The other 95 percent lies in the rigorous system dynamics-driven structuring of models and in the simulation based on these models."*

'Industrial dynamics' (Forrester, 1961) was the first successful SD modelling and simulation project. In fact, SD was originally called 'Industrial dynamics' in this book. In this work, Forrester (1961) defines SD as: *"the investigation of the information-feedback characteristics of systems and the use of models for the design of improved organisational form and guiding policy"*. In the late 1960s and early 1970s, SD technique gained international attention through the work of Forrester (1969, 1971) and the world models of Meadows *et al.* (1972). The technique also expanded the scope of application to traditional academic areas, but with a strong emphasis on socio-economic areas (Coyle, 1977; Roberts 1978; Richardson and Pugh, 1981). Today, the SD technique has an established role in the description and understanding of complex dynamical systems, being often used in non-technical contexts, such as the previously mentioned social or economical systems (Ossimitz, 2001). However, as stated by Pidd (1988), many management scientists are still sceptical of its value. Reasons given by Pidd (1988) to support this argument include: *"that SD is definitely not a highly refined and accurate tool"*.

The SD technique has been closely associated with numerical computer simulation from the outset. The first computer tool based on this technique was DYNAMO, originally developed by Jack Pugh at MIT in the late 1950s and made commercially available from Pugh-Roberts in the early 1960s (Pugh-Roberts Associates, 2003). However, SD tool has been significantly enhanced in recent years by the development of computers, new tools (De Geus, 1988), and alternative control design methods (Mohapatra, 1980; Keloharju, 1983).

2.1.4.3 Hybrid simulation

Hybrid simulation is a combination of the discrete and continuous approaches. Van Beek and Rooda (2000) classify hybrid models as 'Continuous Time Plus' (CT+, or continuous systems extended with discrete elements), 'Discrete Event Plus' (DE+, or discrete systems extended with continuous elements) and CT/DE (equipped with high level language elements in both the continuous and discrete domain). According to them, most of the hybrid languages belong to this category. These models are built using mathematical equations, and enhanced by *"discrete-event additions to enable modelling discontinuities or discrete control actions"*. The capability of combined simulation can be established through a consideration of discrete and continuous approaches. Hence, this approach has not been considered independently and the chosen generic techniques for this category are SD and DES (see Table 5).

2.1.5 Optimisation approaches applied to simulation

Optimisation is not a modelling technique. However, optimisation methods are closely linked to symbolic simulation techniques, especially when a computer tool is used to model them. The calculating power of computers, added to the possibility to create 'batches' of experiments that can be run without the user's intervention, has facilitate the use of optimisation approaches within these modelling techniques. Optimisation can be defined simplistically as *"the action of finding the best solution"* (Klemola and Turunen, 2001). Similarly, Tiwari (2000) defines optimisation as *"the process of selecting a particular design that is feasible and also superior to all other feasible alternative designs, based on some pre-defined criteria, from a set of feasible alternative designs"*.

Optimisation techniques tend to maximise or minimise a global characteristic of a decision process by exploiting certain available degrees of freedom (variables of the system) under a set of restrictions (constraints or boundaries of the system) (Fishman, 1973). There are two ways to obtain an optimum design: through a manual process or by using an algorithmic approach (Roy, 1997). The manual process improves a design by repeated modifications. The design variables are changed one at a time. This task can become too complex to be solved manually if the design involves many variables, especially if variable interaction needs to be considered. In addition, this process is sometimes executed as a trial-and-error exercise, which is both very time-consuming and tedious. On the other hand, the optimisation by using an algorithmic approach can simultaneously determine all the design variables so as to satisfy a set of constraints and optimise a set of objectives. This method is usually quicker, but needs a computable model that includes quantitative data in it.

Techniques such as SD have traditionally relied on the use of intuition and experience for improving system behaviour (Wolstenholme, 1990). However, as Wolstenholme (1990) states: *"this situation is now changing and much effort is being expounded in the development of more formal policy design methods"*.

These formal policies are usually preceded by a number of requisites, such as:

1. An objective function must be defined within the simulation model that summarises overall model behaviour.
2. A number of parameters within the model must be selected as candidates for optimisation, together with a range of feasible numerical values for each.

Each interaction of the procedure starts with a simulation run that calculates the value of the objective function chosen, under the initial conditions chosen for the simulation parameters. Then, the optimisation algorithm treats these parameters as variables for optimisation, and optimises them heuristically; that is, by changing them one at a time using the objective function as a measure of performance. Subsequent iterations repeat this cycle until the optimum value is found or until a predefined value is arrived.

However, many optimisation models have a variety of limitations and problems that a potential user should have in mind. Sterman (1991) cites these problems as: (i) Difficulties with the specification of the objective function, (ii) Unrealistic linearity, (iii) Lack of feedback and (iv) Lack of dynamics.

2.2 The characteristics of Discrete Event Simulation and System Dynamics

Section 2.1 established the concepts of modelling and simulation, and presented a classification of model types and a brief explanation of typical modelling techniques. This section looks deeper into the mechanisms of the two symbolic simulation techniques described in the previous section, namely: (i) Discrete Event Simulation and (ii) System Dynamics.

2.2.1 Discrete Event Simulation

Discrete Event Simulation concerns the modelling of a system as it evolves over time by a representation in which the stated variables change instantaneously at separate points in time (Law and Kelton, 1991), determined by the occurrence of an event (Fishman, 1973). Law and Kelton (1991) define an event as *"an instantaneous occurrence that may change the state of the system"*. Similarly, Kay (1984) defines an event as *"an attempt to change the state of the simulation"*. Fishman (1973) distinguishes these time intervals between two consecutive events as: (i) Random and (ii) Deterministic.

The DES technique is concerned with the 'start time' and 'end time' between changes (denominated previously in this paragraph as events), and not in the time between. Hence, in the DES technique, a change in the system occurs when an event occurs, while the states of entities remain constant between two consecutive events. Due to the dynamic nature of DES simulation (it analyses a system while it evolves over time), this technique requires a time-keeping mechanism to advance the simulated time from one event to another, usually called 'simulation clock' and a list of pending events (or at least the next event) usually called 'event list' (Rubinstein and Melamed, 1998). The research done in order to find efficient event-list manipulation is analysed in detail by Law and Kelton (1991).

A single traffic light is an example of the concept of 'variable time between consecutive events' (Figure 8). In the example shown below, the states of a traffic light are shown based on the DES technique. As can be appreciated in the figure, consecutive events do not occur after the same periods of time. While Δt=15s when changing the state of the traffic light from red to green, it varies to Δt=50s and Δt=100s when changing the state from red to green and green to yellow, respectively.

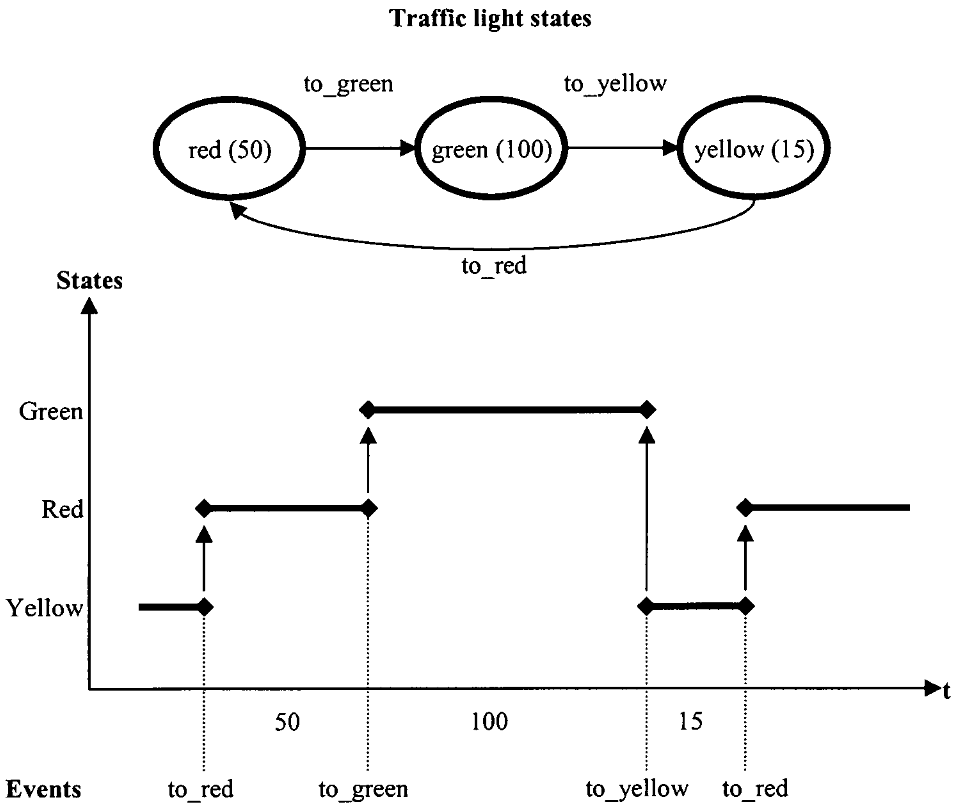


Figure 8: States and events graph in a traffic light using DES

A Discrete Event Simulation model is basically formed by entities (Kay, 1984) and resources (Pidd, 1988). Kay (1984) defines entity as *"an item which changes from one discrete state to another as events occur with the progression of time"*. Pidd (1988) also includes 'resources' as basic systems elements, understanding by resources, *"system elements that are not modelled individually, but treated as countable items whose individual behaviour is not tracked in the computer programme"*. Entities can represent both physical components and imaginary ones (machines, etc.). In addition, 'dummy' entities might be required when constructing a simulation model, in order to hold information not immediately associated with any particular element. Whether a system element is modelled by using an 'entity' or a 'resource' is dependent on the purpose of the simulation (Pidd, 1988). Once the required entities are defined, for example, based on the complexity required, the models are built using those elements. The behaviour of the model will depend not only on the entities themselves, but also on the properties and connections related to them (Oyarbide *et al,* 2000b). Basically, in a DES model, three basic elements can be distinguished, allowing the user to build most common models.

These are as follows:

- **Parts**: These flows through the model. They can represent physical components or imaginary ones.
- **Buffers of queues**: These are places where 'parts' can be held.
- **Machines or activities**: These are used to represent anything that takes 'parts' from somewhere, processes them and sends them on to their next destination.

Each of the basic elements described above contains a set of attributes, understanding by attributes, information attached to the entities that they represent. These attributes can be used for two different purposes: (i) To personalise and differentiate different elements of the same type (also called 'class'), and (ii) To control the behaviour of an entity. As stated previously in this section, DES is based on 'event processing'. Carrie (1988) classifies events into two types, namely: (i) Endogenous (or internal) and (ii) Exogenous (or external). The endogenous events include those that are caused by conditions in the model, such as the completion of an operation. Exogenous events are caused from outside the model, for example, by the arrival of a job from the outside world. Similarly, authors such as Kay (1994), Pidd (1988) and Banks *et al.* (1996) classify the different kinds of events that can appear in a DES as follows:

- **Scheduled events (also named Book-keeping):** They can be predicted in advance. Thus, they occur after a determined period of time.
- **Consequential events (also named Conditional or Co-operative).** May or may not be allowed to take place depending upon the state of simulation because they are triggered by a condition (for example, multiple entities waiting for resources when they are available). Thus, these events are not dependent on the simulation clock but will depend on the states of the entities and resources in the simulation (Pidd, 1988). More than one type of consequential event may occur at any time during a simulation.

The basis of DES is that each activity currently active is checked for its finishing time. Initially, the simulation clock is set to zero and the initial condition(s) and event(s) are defined. Next, the most imminent event is processed and the simulation clock is advanced to its occurrence time. It must be noted that the length of each time step might not be constant, because it is the next event or activities to be completed that define it. At the end of each time step, the appropriate activities are stopped and the corresponding variables are updated, following the previously incorporated decision making philosophy. This approach for advancing the simulation time at slack periods of varying length is called the 'next-event time advance' approach (Rubinstein and Melamed, 1998) or simply 'next-event technique' (Pidd, 1988). In this case, the model is only examined and updated when it is known that a change of state is due (Pidd, 1988). Its running mechanism, extracted from Harrell *et al.* (2000) is shown is Figure 9.

There are different ways in which a model can be organised in a computer programme. Kay (1984) describes two approaches, namely: (i) Two-phase and (ii) Three-Phase. Similarly, Pidd (1988) provides a range of approaches, namely: (i) Three-phase, (ii) Event-based, (iii) Activity-based, and (iv) Process-based; although this author states that a wider range of possibilities exist, including a combination of those described above. Three-Phase appears to be the most implemented approach in simulation computer programmes (Fishman, 1973) and is also the preferential method for authors such as Pidd (1988) and Carrie (1988). The three phases are described by Carrie (1988) as:

- **Phase A**: Advance the clock to the time of the next event.
- **Phase B**: Terminate any activity due to end at this time.
- **Phase C**: Initiate any activities that the conditions in the model now permit.

In the three-phase approach, the 'Start' and 'Stop' of simulation are not linked. The simulation process starts advancing the time, then stops and checks all the elements of the system. Thus, the three phase approach can be slow if there are a lot of events occurring in a short period of time (Aitchison, 1995). More characteristics of this approach, and differences between the approaches described above, can be found in literature (Kay, 1984; Pidd, 1988).

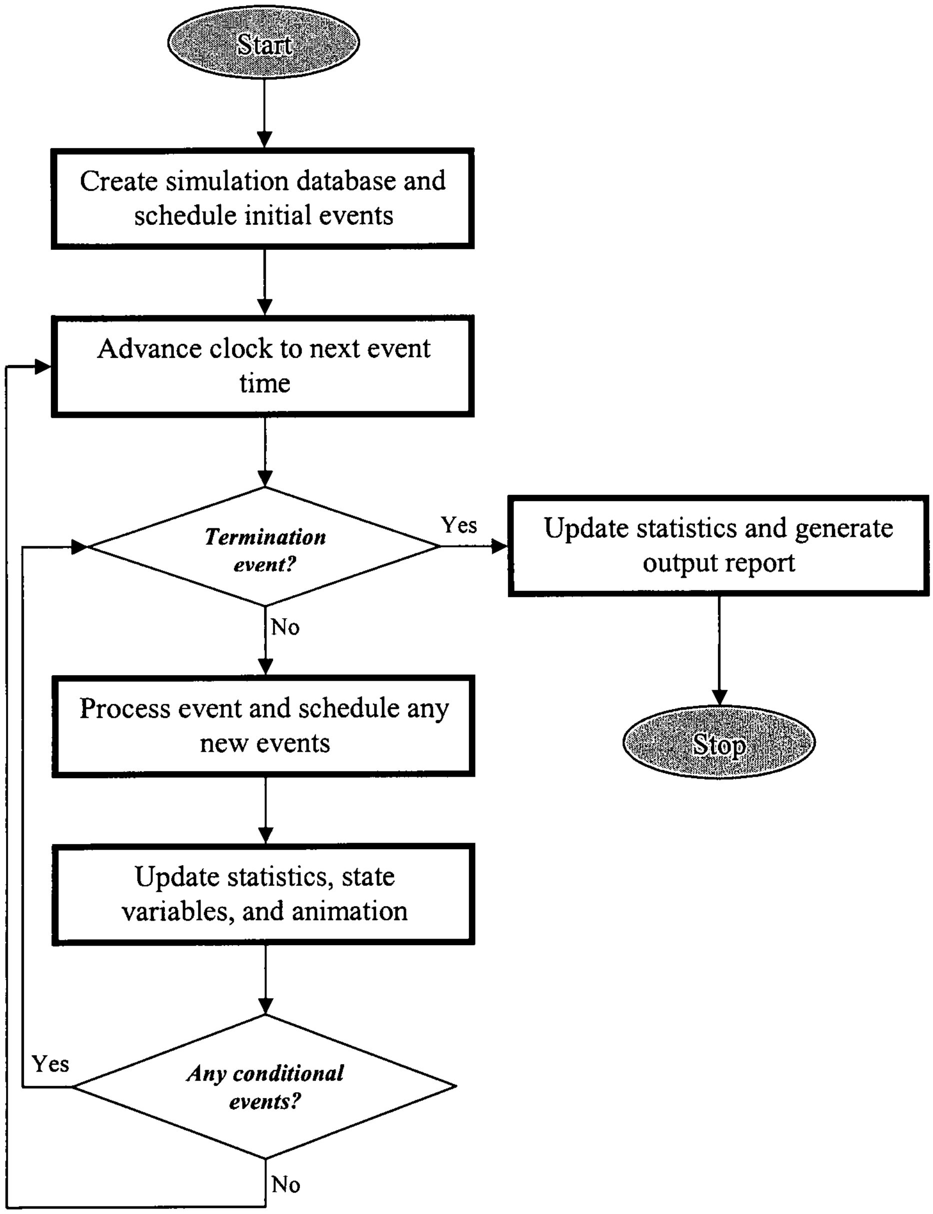


Figure 9: Diagram of Discrete Event Simulation process (Harrell et al., 2000)

2.2.1.1 Stochastic systems and Discrete Event Simulation

DES is closely related to the simulation of stochastic systems. Section 2.3.1 described the importance of manufacturing systems and highlighted the complexity and variability of these processes. When simulating real-world phenomena, there are few situations where the actions of the entities within the system under study can be completely predicted in advance (Banks *et al.*, 1996). Furthermore, in most real cases, there are aspects of a system that are not known in sufficient detail, such as external influences or human factors (Carrie, 1988).

Within the context of manufacturing system design, many examples of causes of variation can be found. For example, the time that an operator requires to repair a machine is often not predictable in advance. However, the continued effort in the selection of appropriate statistical distributions may well estimate some parameters that initially look unknown. For concrete examples of statistical distributions that can be applied in simulation models, refer to Banks *et al.* (1996) or Law and Kelton (1991), who describe and analyse in detail both discrete distributions, such as: (i) Bernoulli, (ii) Binomial, (iii) Geometric and (iv) Poisson; and continuous distributions, such as: (i) Uniform, (ii) Exponential, (iii) Gamma, (iv) Erlang, (v) Weibull and (vi) Triangular.

In practice, simulation packages tend to generate pseudo-random numbers in order to select a particular value from a given distribution (Carrie, 1988). The generation of pseudo-random numbers however, must ensure that these are uniform and independent (Banks *et al.* 1996). Although several methods do exist to generate pseudo-random numbers (Law and Kelton, 1991), the lineal congruential method is the most widely used technique for generating random numbers (Banks *et al.*, 1996). This method, initially proposed by Lehmer (1951) (cited in Banks *et al,* 1996) produces a sequence of integers, X_1, X_2, ... between zero and (m-1) according to the following recursive relationship:

- $X_{i+1} = (aX_i + c) \mod m$; where $i = 0, 1, 2, \ldots$

The initial value X_0 is called the 'seed', 'a' is called the constant multiplier, 'c' is the increment, and 'm' is the modulus. If '$c \neq 0$', the form is called the 'mixed congruential method', while if '$c = 0$' the form is known as the 'multiplicative congruential method'. As explained by Banks *et al.* (1996), the selection of the values for 'a', 'c', 'm' and 'X_0' drastically affects the statistical properties and the cycle length.

There are some scenarios, however, where the application of known distributions (see examples provided above) might be impossible or unnecessary. In those cases, if data are available, an empirical distribution (using, for example, data based on previous experiences) can be used. Whether or not this method is better than using theoretical distributions depends on different factors, such as the time required for collecting the empirical data, accuracy, etc.

2.2.1.2 Activity cycle diagrams

Activity cycle diagrams (or entity life cycle diagrams) are one way of modelling the interactions of the entities and are particularly useful for systems with a strong queuing structure (Pidd, 1988). They were popularised in the 1970s and are normally associated with the activity and three-phase approaches described later in this chapter (Hills, 1971; cited in Pidd, 1988). Activity cycle diagrams make use of only two symbols (Figure 10) to show the entities of a model and their interactions.

As Pidd (1988) states:

> *"Each entity is considered to have a life cycle which consists of a series of states. The entities move from state to state as their life proceeds".*

The way in which two symbols are used to draw an activity cycle diagram is described by Carrie (1988), who specifies five conventions for the appropriate use of this type of diagram. These are:

1. Each type of entity has an activity cycle.
2. The cycle consists of activities and queues.
3. Activities and queues alternate in the cycle.
4. The cycle is closed.
5. Activities are depicted by rectangles and queues by circles or ellipses.

In addition, Carrie (1988) mentions that there are two basic forms of activity cycle, concerned with the case where the entity either:

1. May perform one or more different activities in any sequence or is idle.
2. Must perform activities in a definite sequence.

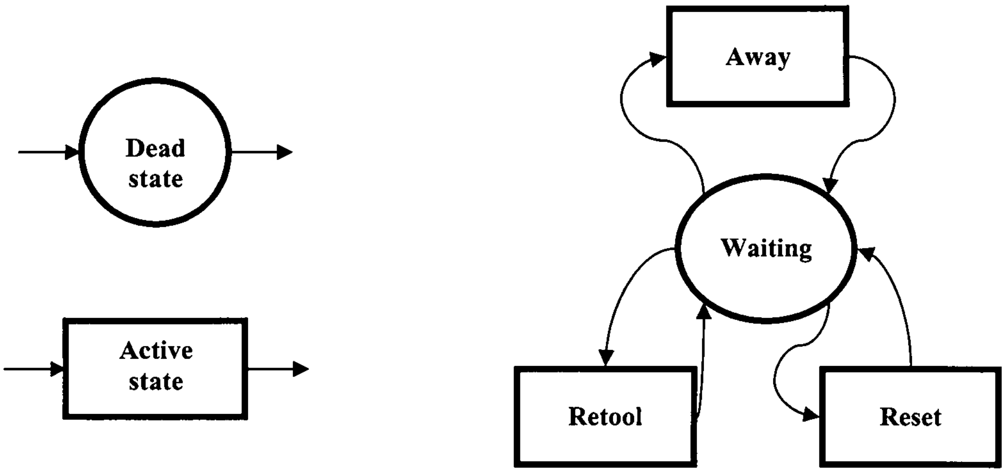


Figure 10: a) Symbols for activity cycle diagrams and b) an example of a diagram (Pidd, 1988)

The example provided in Figure 10 represents an operator of a simple machine, where the operator has to perform the following two tasks: (i) Reset the machine and (ii) Retool the machines, under certain conditions. In addition, he or she may be unavailable while attending to personal needs. Thus, the active states are: (i) Away, (ii) Retool and (iii) Reset, while the dead state is represented by: (i) Waiting, for those instances where the operator is unavailable.

Although activity cycle diagrams are useful in a large variety of systems, and have been successfully implemented in commercial software such as the HOCUS (Hand or Computer Universal Simulator) package (P-E International, 2003), there are systems which do not easily fit this notation (Pidd, 1988). Examples of these are systems where the interruption of an active state may occur before it reaches its scheduled termination. For more information, read Chwif *et al.* (1999), who compare the Activity Cycle Diagram approach with the Condition Specification approach and provide a number of advantages and disadvantages for each.

2.2.2 System Dynamics

SD technique is concerned with constructing a model in which the stated variables change continuously with respect to time (Law and Kelton, 1991). It is rooted in the engineering traditions of control theory and feedback analysis (Alfeld, 1994), emphasising system structure rather than data collection. In addition, it also focuses on interdependencies, feedback effects, time dependencies, and causality in the object that is being represented (Ellis, 1998). Thus, while many traditional modelling techniques apply statistical tools to data sets, and infer causal relationships between correlated variables, SD develops explicit descriptions of causal relationships within a formal feedback structure.

In the field of SD, a 'system' is defined as a collection of elements that continually interact over time to form a unified whole, while the term 'dynamics' refers to change over time (Martin, 1997a). One example of a system is an assembly line. The structure of an assembly line is defined, for example, by the interactions between quantities of raw material, stock levels, production of goods and control policies; whereas the behaviour is due to the influences of raw materials, availability of machines and operators, and environment. One feature that is common to all systems is that its structure determines its behaviour (Martin, 1997a). The selection of an appropriate level of detail, problem boundaries, and similar considerations constitute the 'art' aspect of dynamic simulation model development (Shreckengost, 1985). An advantage of the SD technique with respect to alternatives techniques such as DES is that, in SD, the complexity of the model increases linearly with respect to the complexity of the system (Khurana, 1999).

Thus, SD is a technique that aims to enhance learning in complex systems. The purpose in applying SD is to facilitate understanding of the relationship between the behaviour of a system over time and its underlying structure and strategies/policies/decision rules (Wolstenholme, 1990). Thus, the mechanisms applied by this technique are grounded in the theory of non-linear dynamics and feedback control developed in mathematics, physics and engineering (Sterman, 2000). Feedback structures (explained later in this section) are an essential part of models based on the SD technique. As Sterman (2000) states:

> *"Much of the art of SD modelling is discovering and representing the feedback processes, which, along with stock and flow structures, time delays, and nonlinearities, determine the dynamics of a system".*

A SD model is basically formed using the three basic elements of this technique (Figure 11): (i) Levels, (ii) Rates and (iii) Converters (Wolstenholme, 1990). The SD technique considers 'levels' as accumulations; they hold the current state of the system. In the words of Martin (1997c), levels represent *"What you would see if you were to take a snapshot of the system"*. On the other hand, rates are considered as the elements that do the changing operation, increasing or decreasing the levels over the time.

- **Level:** Levels represent the state of some element of the system at the point in time that the measurement is taken. Levels are expressed in units (such as 'Engines'), rather than in units per time (such as 'Engines per day'). 'Engines' is an example of a stock variable, because it represents the state of something being measured at a particular moment in time.

- **Rate:** Rates represent the change of some element of the system across a specified time interval. The value of a rate at a particular point in time represents the amount by which it will modify the stock variable associated with it during the next Delta Time (DT) period. Rates are expressed in units per unit time. 'Assembly Cycle Time (ACT)' is an example of a flow variable; its rate equivalent would be 1/ACT. If the rate ACT is associated with the level 'Engines', and the value of the ACT at the current DT is 5, this implies that ACT will increase the value of 'Engines' by 5 in the next time interval.

- **Converter:** Converters are 'intermediate' variables, which relate levels to rates, or rates to other rates. They are used as information links through the system, improving the decision making task.

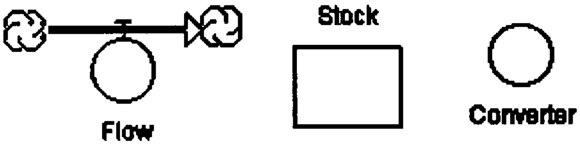


Figure 11: Typical representation of SD basic elements. a) Flow or rate, b) Stock or level, c) Converter

In addition to the basic elements and integration mechanisms described above, SD is also based on some basic principles, which make the technique structured and easy to understand. To make SD a technique capable of being applied successfully to a wide range of applications, Forrester (1969) developed the 'System Principles', or core principles on which SD is based on. These are enumerated and explained in Appendix A and summarised below:

1. The 'feedback loop' is the basic structural element of the system.
2. 'Levels' and 'rates' are fundamental to loop substructure.
3. 'Levels' are accumulations (integration).
4. 'Levels' are changed only by the 'rates'.
5. 'Rates' depend only on 'levels' and 'constants'.

6. Every equation must have dimensional equality.
7. 'Levels' completely describe the system condition.
8. Information links connect 'levels' to 'rates'.

2.2.2.1 An overview of Qualitative and Quantitative System Dynamics

Wolstenholme (1990) points out that the SD technique can be applied using two different approaches, namely: (i) Qualitative and (ii) Quantitative and states that the election between these two types is based on the identification of a problem or cause for concern. While the first approach is concerned with creating cause and effect diagrams, the second approach involves deriving the shape of relationships between all variables within the diagrams, the calibration of parameters and the construction of simulation equations and experiments (Wolstenholme, 1990). A summary of the scope and typical objectives of both qualitative and quantitative SD is provided in Table 8.

QUALITATIVE SD *Diagram construction and analysis phase*	QUANTITATIVE SD *Simulation phase*
To create and examine feedback loop structure of systems using resource flows, represented by level and rate variables and information flows, represented by auxiliary variables. To provide a qualitative assessment of the relationships between system processes (including delays), information, organisational boundaries and strategy. To estimate system behaviour and to postulate strategy design changes to improve behaviour.	To examine the quantitative behaviour of all system variables over time. To examine the validity and sensitivity of system behaviour to changes in: (i) Information structure, (ii) Strategies and (iii) Delays/uncertainties. To design alternative system structures and control strategies based on: (i) Intuitive ideas, (ii) Control theory analogies and (iii) Control theory algorithms, in terms of non-optimising robust policy design. To optimise the behaviour of specific system variables.

Table 8: Quantitative and Qualitative SD. A subject summary (after Wolstenholme, 1990)

When SD is used qualitatively, it is concerned with creating 'cause and effect' diagrams or system maps (known as casual loop or influence diagrams) according to pre-established rules. These types of models are often used to explore and analyse the system (Wolstenholme, 1990). Considerable thought has been given to improving the ease of conceptualisation of SD models in last decades (Morecroft, 1982), with the aim of guiding the discussions of team members. Although the qualitative nature of these models mean that comprehensive simulation is not possible, it is sometimes possible sometimes to estimate the general behaviour of the system by studying the feedback loop structures of the model.

On the other hand, SD can be used quantitatively. This is the more conventional and traditional use of SD, and it involves deriving the shape of relationships between all variables within the model, the calibration of parameters and the construction of simulation equations and experiments (Wolstenholme, 1990). In particular, quantitative SD has its roots in systems of differential and difference equations (Forrester, 1980). Thus, according to Pidd (1988), a quantitative SD model is *"a set of difference equations whose variables change their value through time"*. Forrester (1969) developed this approach partially based on the analogy between physical control systems and the control systems employed in organisations (Pidd, 1988).

The typical differential equation has the form:

- $\dot{x}(t) = \frac{dx}{dt}$; or: $x = \int_{T1}^{T2} f \cdot dt$; graphically, this can be represented as:

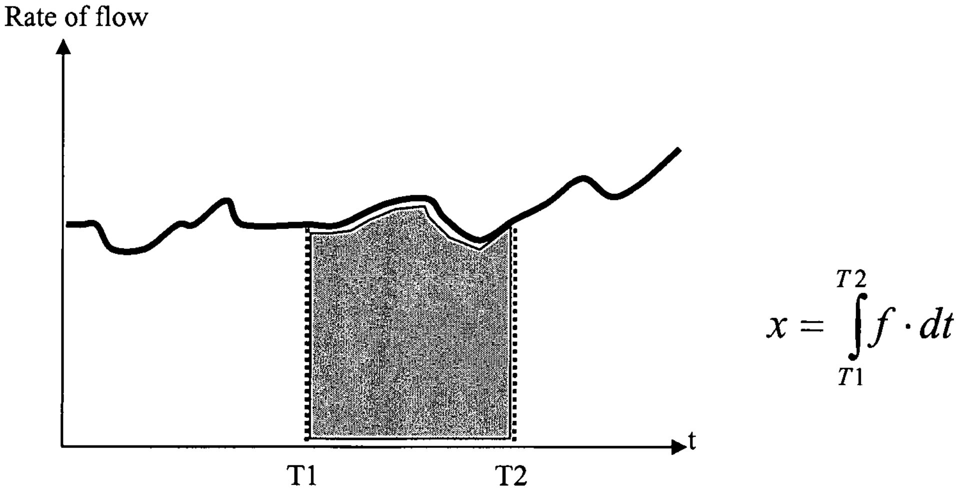


Figure 12: Graphical representation of the differential equations (after Carrie, 1988)

Within manufacturing systems, an example of a typical differential equation could be:

- $WIP = Initial + \int_{0}^{T} production(t) \cdot dt$

  where:

  - 'WIP' = Work in progress of a particular element in the time 'T'.
  - 'Initial' = Initial value of the 'WIP' of that element.
  - 'production(t)' = Function that represents the shape of the productivity rate over time.
  - 't' = Time.

Pidd (1988) however, identifies a problem associated with this approach. Although differential equations can often be written to model the dynamic behaviour of a system, in many cases these sets of equations cannot be integrated directly. This view is also supported by Law and Kelton (1991) who state that analytical solutions are not possible for most continuous models. In addition, the resolution of differential equations by computer methods might be a difficult task, requiring highly complex algorithms and, as stated previously are only effective with those differential equations that can be solved. Thus, Forrester (1969) provides a simplified version of numerical integrations to be applied in SD models using a time-slicing approach based in first order difference equations (Pidd, 1988), applying techniques such as 'Runge-Kutta' or 'Rectangular methods' to obtain numerical results (Law and Kelton, 1991). A typical difference equation has the form:

- $x_{t+\Delta t} = x_t + f(x_t; \vartheta)$

  where:

  $\Delta t = (T+1) - (T)$

Similar to differential equations, a manufacturing analogy can be made. Thus, an example of difference equations within manufacturing could be:

- $WIP_k = WIP_j + \Delta t \cdot (FlowIn_{jk} - FlowOut_{jk})$

  where:

  - 'WIP_k' = Work in progress of a particular element measured in the time 'k'.

— 'WIP_j' = Work in progress of a particular element measured just an instant before, called 'j'.

— 'Δt' = Time different between the instants 'k' and 'j'

— '$FlowIn_{jk}$' = Amount of components that have entered into the element during the time 'Δt'.

— '$FlowOut_{jk}$' = Amount of components that have leaved the element during the time 'Δt'.

As can be appreciated from the previous equations, the SD technique relies on two main types of equations, namely: (i) Level equations and (ii) Rate equations (Pidd, 1988). Thus, to calculate the value of the levels within a model, in addition to the levels equations, it is only necessary to provide the initial values of each level of the model and the equations for each flow. The difference equations approach provides a number of advantages with regard to differential equations, such as: (i) They can all be solved and (ii) The algorithm for implementation in computer tools is simpler and faster. However, it must be noted that difference equations are a simplification of differential equations that can only be considered valid if the time interval between two consecutive calculations (Δt) is 'infinitesimal' (Gilbert and Troitzsch, 1998). Within the SD technique, 'Δt' is called 'delta time' (DT) (expanded in Section 2.2.2.4) and the value of this time increment is chosen according to the characteristics of the model and the analysis that is going to be undertaken.

2.2.2.2 Delays and non-linearities

Delays are inherent in many management processes. Their presence usually makes it more difficult to control the system in hand (Pidd, 1988). Different sources of delays and a variety of systems responses to typical delays are described by Fishman (1973). The simplest type of delay is the one produced into the flows. This delay appears for example when one machine finishes its operations and the next one cannot start because there is a conveyor in between. Pidd (1988) divides the most common types of delays into three categories, namely:

- **Exponential delays**: These occur when part of a system takes some time to respond to changes in its input.
- **Pipeline delays**: These are an analogy with a pipeline of known length into which material is fed and from which it flows once the material has passed through the pipe.

- **Batch delays**: These are analogous to ovens in which batches of items are placed, cooked for an interval, and then all released after some time interval.

In addition to delays, real systems often include non-linearities. One process is considered as linear if its response is proportional to the stimulus given to it. The simplest models are the linear ones, because their mathematics are relatively straightforward and can adequately represent the behaviour of many realistic processes over a useful range of conditions. In manufacturing companies, there can also be non-linear situations, for example, the quality of the work produced against the number of working hours. In SD, the best way to solve these non-linearities is to create IF-THEN-ELSE situations or to define tables of situations. This solution decomposes the non-linear problem into a number of parts in which the problem can be considered as linear.

2.2.2.3 Feedbacks and System Dynamics diagrams

Feedbacks are an essential part of SD models (Kirkwood, 1998; Wolstenholme, 1990). In addition, Zhu (1996) states that *"feedback in systems causes nearly all the dynamic behaviour"*. They are defined by Martin (1997b) as: *"a process whereby an initial cause ripples through a chain of causation ultimately to re-affect itself"*. Coyle (1996) gives another more formal definition of feedback loops. He defines feedback loops as *"a closed chain of cause and effect in which information about the results of actions is fed back to generate further action"*.

When modelling real world scenarios, the complexity of the model is often raised from the interactions (feedbacks) amongst the components of the system rather than the complexity of the components themselves (Sterman, 2000). Production plants based on 'functional lay-out' with generic machines can be considered as examples, with complex control policies and set-up operations.

As stated by Coyle (1996), feedback loops generate further action. To do that, it is necessary that feedback loops include at least one rate and one level (Wolstenholme, 1990) in order to integrate the first into the second and produce behaviour. Feedback loops can be divided into two main groups, namely: (i) Positive (or self-reinforcing) and (ii) Negative (or self-correcting). While positive loops tend to reinforce or amplify whatever is happening in the system, negative loops counteract and oppose change (Sterman, 2000). It must be noted, however, that positive feedbacks need to be combined with negative feedbacks, due to the impossibility in practice to grow real quantities forever. For more information, refer to Albin (1996) and Albin and Choudhary (1996) who analyse in detail different scenarios of both positive and negative first-order feedback loops, Coronado (1996) who analyses the effect of constant flows into the system and Stanley and Zhu (1996), who analyse the more complex situations in which combining feedbacks are present.

Figure 13 and Figure 14 exhibit examples of negative and positive feedbacks, respectively. In the first figure (negative feedback), the actual level of production is compared with the target, and the difference is used to govern the rate speed of a machine. As can be appreciated in Figure 13, at least two different behaviours might occur depending on the existence (or not) of delays within the loop. In an 'undelayed loop' situation, the gradient of the curve is dependent on the capacity of the machine and the difference between the desired level and the real one. However, the real state of the level never overcomes the targeted one. On the contrary, in a 'delayed loop' situation, the inventory continues increasing although the ordering is stopped when inventory reaches its target. This is due to the delay between the production ordering and production completion. The second figure (positive feedback) is an example of feedback loops that grown indefinitely rather than target seeking. In the example provided, as population increases, birth rate also increases, which causes population to increase faster, and so on.

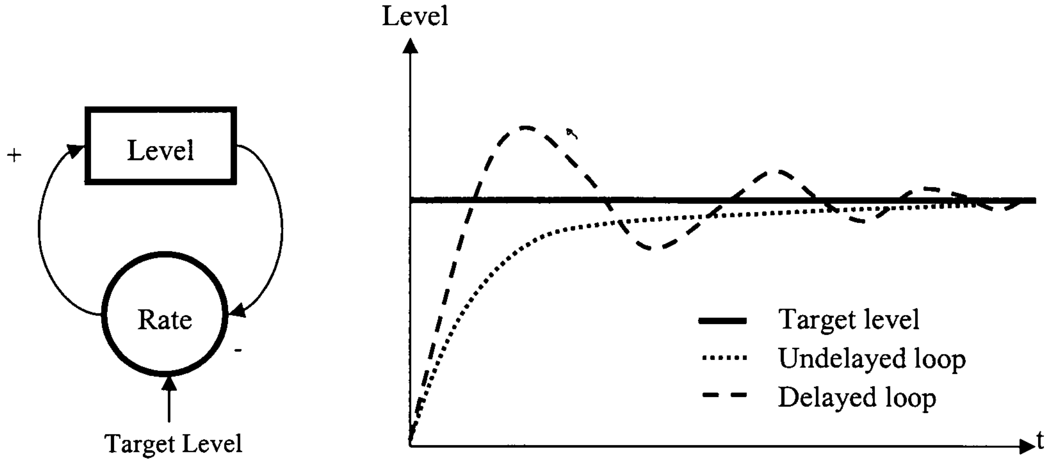


Figure 13: Example of a negative feedback loop and its classical mode of behaviour (after Sterman, 2000 and Wolstenholme, 1990)

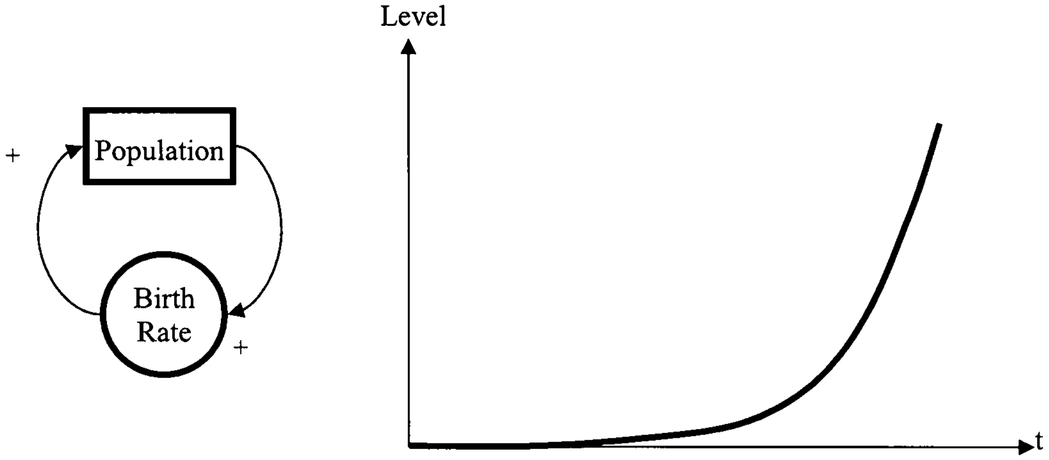


Figure 14: Example of a positive feedback loop and its classical mode of behaviour (after Sterman, 2000 and Wolstenholme, 1990)

The notation used to represent feedback loops in Figure 13 and Figure 14 is called ‘causal-loop diagram’. Within this notation, the direction of the arrow head indicates causality, whereas the sign at the arrow head indicates the effect of causality (positive or negative). This form of notation, however, is not the most popular for representing SD models, because it exhibits a variety of problems when used with complex models. Richardson and Pugh (1981) list a number of problems associated with ‘causal-loop’ diagrams. The main ones are as follow:

1. Causal-loop diagrams obscure the stock and flow structure of systems.
2. They do not make distinction between information links and rate-to-level links.

Fortunately, an alternative notation to model SD models does exist. It was also created by Forrester (1961) and is commonly called ‘levels and rates’ diagrams but it is also known as ‘stock and flow’ diagrams. It consists mainly of three different types of elements: (i) Levels (or Stocks), (ii) Rates (or Flows) and (iii) Information; but includes some variations in the symbols depending upon the nature of the element. For example, within this notation, material flows are represented differently than information flows. This graphical notation also hints at the differences between levels and rates. The rectangular boxes (levels) look like containers and their connection (rates) looks like a pipe with a valve. This notation has a wider acceptance within users of the SD technique, and it is also implemented in the most important SD based tools, such as Stella/iThink. An example of a simple SD diagram based on this notation is shown in

Figure 15. In this example, the production rate will increase the inventory level, based on the desired productivity and the available labour.

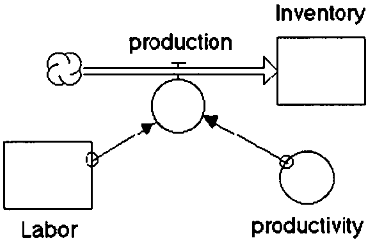


Figure 15: An example of 'Levels and Rates' diagram

2.2.2.4 Delta Time (DT)

Time handling in SD is a time slicing method (Pidd, 1988). Thus, the key factor involved in the calculation of results when using a continuous modelling technique, such as SD, is the Delta Time (DT), or time interval between calculations. DT is defined as the period of time between two consecutive simulation stops and calculations. Therefore, DT defines the resolution of the simulation. A high value of DT means that simulation can be inaccurate but fast to run, and a low value of DT means the opposite. Selection of DT can make an important difference in the behaviour of a model (Coyle, 1977). Forrester (1961) suggests that the time increment should not exceed DEL/2D, where 'DEL' is the total length of the highest order delay and 'D' is the order of the delay. Thus, the higher the order of the delays of a system, the smaller DT has to be. Barton and Tobias (1998) compile several suggestions in order to select a correct value for DT. As a summary (Zaraza, 1998), states:

> *"In general use the longest DT possible that allows accurate model behaviour. Whenever possible, use a DT that fits the sequence 1/(2^n)".*

Zaraza (1998) provides a practical method for selecting an appropriate value of DT for a given model. This method starts by analysing the delays caused by feedback loops. The first stage in choosing DT is to set DT to half of the value of the shortest delay of the model, or set DT to be equal to one in case of non-existence of delays. This DT provides a starting point. Then, run the model. To see whether or not a shorter DT is necessary, change the DT to half the previous value and run the model again. Compare the results, examining both the shape of the graph and a table of numerical results. In particular, it is convenient to look for significant changes in behaviour (does an oscillation disappear? are rates of growth drastically different? do stocks that were negative in the previous run now remain positive?) If any of these conditions occur, the previous DT may have been too long. The current DT may still be long. The next step consists of reducing DT by half again and re-running the model. These steps must be repeated until similar results are obtained in two successive runs. At that point it is likely that behaviours observed are the result of the actual model and not an artefact of the choice of DT. Use the highest DT of the pair when running the model.

A general rule of thumb in choosing DT is represented in Figure 16. This figure has been developed under the guidelines shown in Zaraza, (1998):

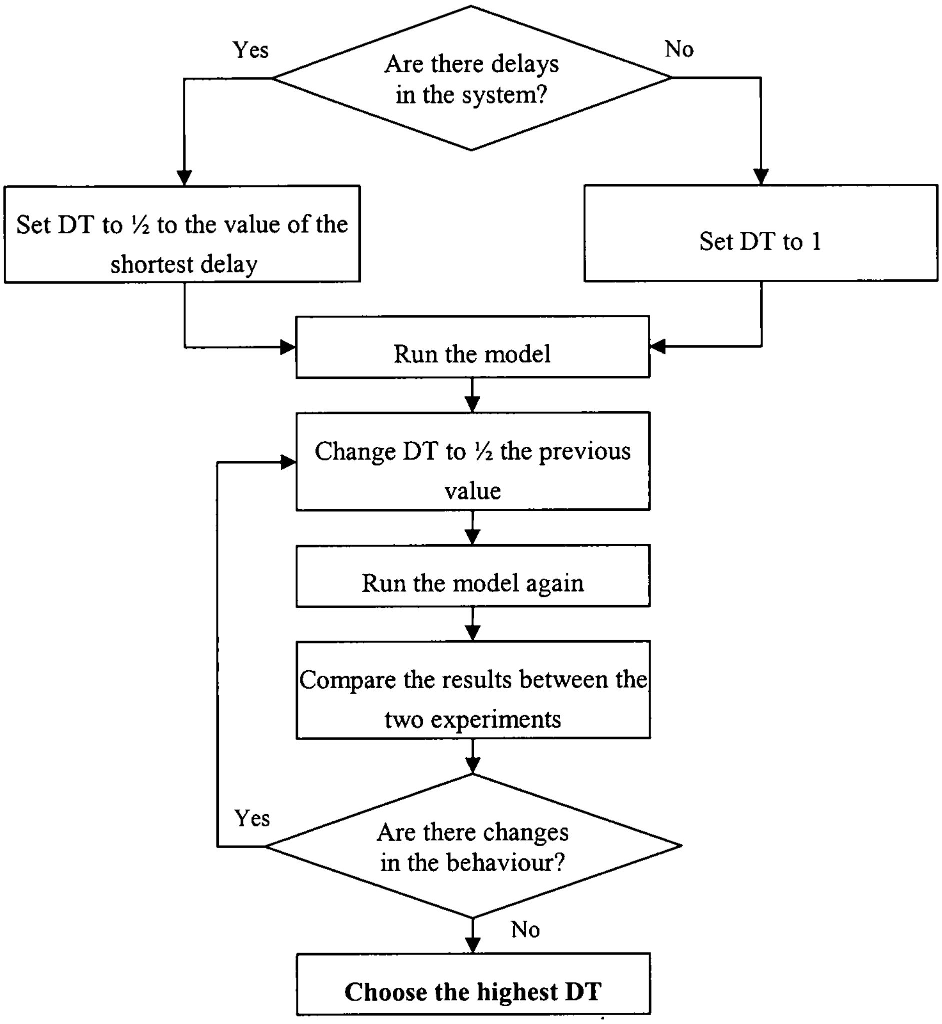


Figure 16: General rule in choosing DT

2.3 Importance of simulation techniques in industry

This section establishes the importance of manufacturing to the global economy (Section 2.3.1). It also provides an overview of the manufacturing system design process (Section 2.3.2), and the importance of simulation techniques as evaluation tools within this process (Section 2.3.3).

2.3.1 Manufacturing industry and its environment

Manufacturing is by far the largest single contributor to the global economy, accounting for almost three-quarters of the world's trade (Scheele, 2000). Within the UK, manufacturing accounts for over one fifth of total national output (Gross Domestic Product or GDP) (Online Learning Resource, 2003). Within the North East of the UK alone, over 5000 manufacturing enterprises contribute £6bn to the regional economy, employ over 150,000 people and generate exports of goods worth almost £7bn (Department of Trade and Industry, 2002). It is also widely accepted that economic growth is largely dependent upon the ability to continually manufacture existing and new products that meet current and future market requirements. The contribution of the manufacturing sector to the wider economy is a topic of hot debate across most of the developed world (Nellis and Figueira, 2003). Further, many economists argue that the economic health of manufacturing has important implications for other industries. However, the cost of creating a manufacturing system is also high. For example, Ford invested £750 million at their Bridgend plant for the engine manufacturing system of the Zetec engine (Aitchison, 1995).

Askin and Standridge (1993) view the purpose of manufacturing, at least idealistically, as *"the enrichment of society through the production of functionally desirable, aesthetically pleasing, environmentally safe, economically affordable, highly reliable, top-quality products"*. Increasing competition and globalisation has forced industry to not only increase the quality of their products, but to also produce them at minimum cost (Chang and Wong, 2002; Helliwell, 2002). In today's worldwide marketplace, industry is learning that constant improvement is a prerequisite for continued existence.

According to Baudoin (1995) *"the scope of manufacturing systems entails an extensive range of subject areas related to the entire lifecycle of products, from development through production and beyond to product support and re-manufacturing or recycling"*. The activities of the manufacturing systems will attempt to address all of those aspects of manufacturing. In this context, simulation seems to be more suited to the manufacturing system design and production stages. In the design stage, simulation can aid practitioners to identify levels of inventory, throughput values, etc. On the other hand, in the production stage, simulation can be used to test different policies (e.g. batch sizes) and analyse the effect on the whole system.

Manufacturing, or the process of transformation, is related to the operations required to produce goods, services or a combination of the two (Slack *et al.*, 1998). By transformation, Slack *et al.* (1998) mean that the manufacturing process uses some 'resources' to change the state or condition of something to produce 'outputs'. In other words, manufacturing operations take in a set of input resources, use them either to transform something, or to be transformed themselves, into outputs or goods and services. These inputs can be classified as either: (i) Transformed resources or (ii) Transforming resources (Slack *et al.*, 1998). The first type involves the resources that are treated in some way (materials, etc.), while the second type contains those resources that act upon the transformed resources (facilities, staff, etc.). Figure 17 illustrates the key operations related to the manufacturing activity.

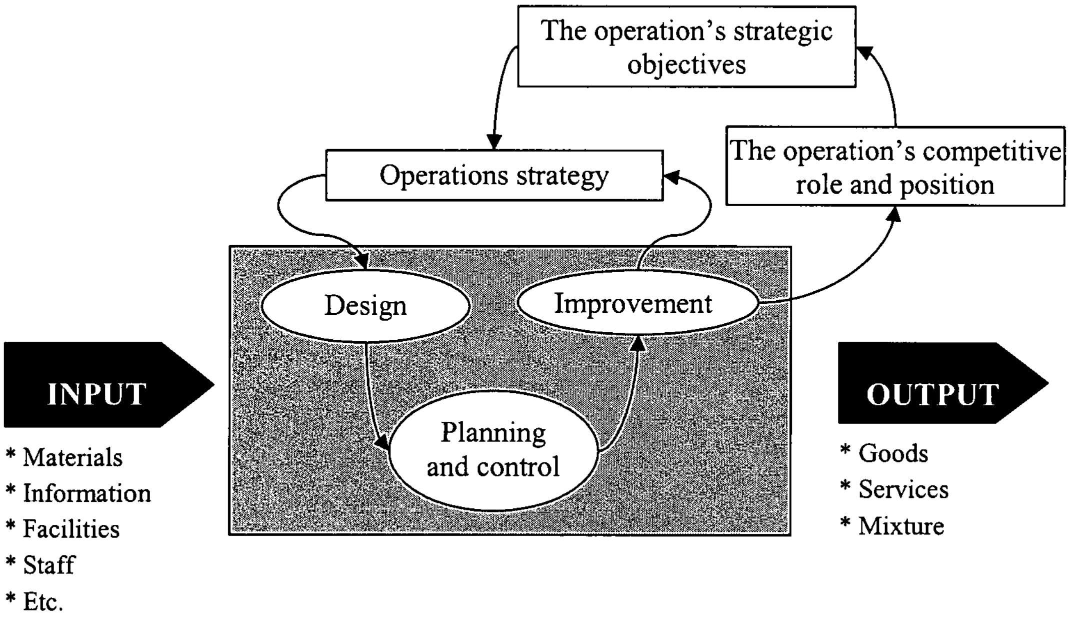


Figure 17: A general model of operations management (after Slack et al., 1998)

The turbulent environment in which most organisations do business means that the operations function has to continually adjust to changing circumstances (Slack *et al.*, 1998). It seems apparent that a company's performance depends on how well its manufacturing capabilities match the environment, including not only the technical specifications but other aspects such as delivery time, cost, flexibility, etc (Krajewski and Ritzman, 1999). Over the years, the manufacturing industry has been a major source of rapid productivity growth (Department of Trade Industry, 2002). However, companies cannot always evolve as quickly as customer demand, and frequently manufacturing operations are not updated as often as the volatile market environment necessitates (Okudan and Kabadayi, 2001).

Companies make a substantial effort to minimize 'environmental' disruption. At the manufacturing level, one way to achieve this is by buffering or insulating the operations function from the external environment (Slack *et al.*, 1998). This can be done through physical buffering or organizational buffering, although this practice is not always advisable because it usually increases the complexity of the manufacturing systems.

Companies such as Ford Motor Company align their manufacturing operations for continuous environmental improvement by creating systems such as the Ford Production System (FPS) to continually monitor and adjust the processes to the requirements (Ford Motor Company, 2002).

2.3.2 Manufacturing system design process

Section 2.3.1 outlined the importance of the manufacturing industry and the necessity for manufacturers to adapt their systems in order to maintain or gain a competitive advantage. It therefore seems logical that the process of manufacturing system design is a key activity within many companies.

Design can be considered to represent a process that begins with recognition of the need and the conception of an idea to meet this need. Thus, in design decision making, the main aim of the designer is to identify a solution that meets or closely meets the performance requirements of the design, while satisfying all the constraints (Roy, 1997). Manufacturing design is closely related to the design of the product that is to be manufactured. Concurrent design practices recommend overlapping the design of the product with the design of the manufacturing facilities in order to reduce: (i) Costs (for example, by choosing appropriate materials or easy of manufacture) and (ii) Development time, since the initial concept until the product reaches the market is achieved more quickly (Suh, 1990).

In practice, the manufacturing systems design activity tries to determine the resources and their configuration to support the organisation's objectives, whereas typical objectives in a product design process might be; obtaining the expected specifications, analysing ease of assembly, etc. However, the concepts applied in the process of designing a product can be valuable for designing manufacturing systems.

Within the context of layout planning, design involves decisions about the quantity, type and configuration of the facilities that need to be included, as well as the location and quantification of the space and capacity required between them (Krajewski and Ritzman, 1999). When designing or redesigning new manufacturing facilities, some key objectives usually need to be satisfied. The selection of appropriate performance measures that reflect these objectives is a key task in the design process. Table 9 provides a categorisation of some manufacturing objectives and typical performance measures. They are a result of a study conducted by Okudan and Kabadayi (2001), which involved the participation of more than 60 companies and are sorted in order of importance. Similarly, Law and Kelton (1991) include several common measures of performance obtained for a simulation study of a manufacturing system, such as: (i) Throughput, (ii) Time in system, (iii) Sizes of in-process inventories (WIP, queue sizes), (iv) Utilisation of equipment, (v) Proportion of time that a machine is broken, waiting for parts, blocked or undergoing preventive maintenance.

Manufacturing objectives	%
Improving quality	18.1
Reducing production cost	16.8
Increasing production volume	13.4
Increasing labour productivity	12.8
Increasing market share	12.1
Reducing machine set up times	7.4
Increasing delivery reliability	6.7
Increasing profitability	4.7
Reducing overhead cost	4.0
Reducing delivery time	4.0

Selected performance measures	%
Scrap and rework	22.9
Productivity	18.8
Direct labour cost	12.5
Capacity utilization level	11.8
Non-value adding time	9.7
Percentage of raw materials in production	9.7
Delivery time	8.3
Machine set up time	6.3

Table 9: a) Manufacturing objectives and b) Performance measures (Okudan and Kabadayi, 2001)

Once objectives have been set, the manufacturing system design process can follow a set of formalised steps. This is a complex task and is crucial to the future of a company

(Chan and Jian, 1999). The manufacturing design process has been discussed in detail by authors such as Compton (1988), Suh (1990), Tempelmeier and Kuhn (1993), Sule (1994), Wu (1994), Heragu (1997), Meyers and Stephens (2000), and Wu (2000). An example of a typical process for manufacturing system design has been outlined in Chapter 1 (see Figure 1). Morris (1977) argues that there are typically four main steps in the decision-making process. He also states that *"the success of the decision is strongly dependent on the sequential execution of these stages"*.

Similarly, Slack *et al.* (1998) decompose this process into five steps, as illustrated in Figure 18, and as described below:

- **Concept generation:** Developed from ideas generated both inside (staff, R&D department, etc.) and outside the organisation (customers, competitors, etc.). It is important to find broad, general concepts. Specific aspects will be discovered in the following steps.
- **Concept screening:** Concept screening analyses the feasibility, acceptability and risks of the previous concepts (considering, for example, marketing, operations or financial issues) and decides which ones will be capable of further development. A good definition of the objective can save time and effort, and therefore money. Details that tend to make the process more difficult to solve can be analysed in depth. Each detail can, in some cases, be a constraint. All constraints must be analysed in detail in this stage, separating the affordable solutions from the unfeasible or impossible ones. It is also useful to detail all consequences that a partial solution can cause, to check its compatibility with other partial solutions.
- **Preliminary design:** Once an acceptable concept has been generated, a preliminary design is created in order to detail the product's components and define its manufacturing process.
- **Evaluation and improvement:** The purpose of this stage in the design activity is to take the preliminary design and determine whether it can be improved before the product is tested in the market. For many problems it is possible to consider several alternative solutions rather than have a 'best' one. The selection of the best solution must be based on real considerations, which means there can be a solution that solves the problem better, but time or money constraints prevent its implementation.

- **Prototyping and final design:** The improved design is turned into a prototype so that it can be tested.

From the stages described in the previous paragraph, it can be noted that design involves progressively reducing the number of possibilities until the final design is reached. While a relatively high number of concepts might be discussed within the conceptual stage, these alternatives need to be narrowed down before the final design to reduce cost and development time.

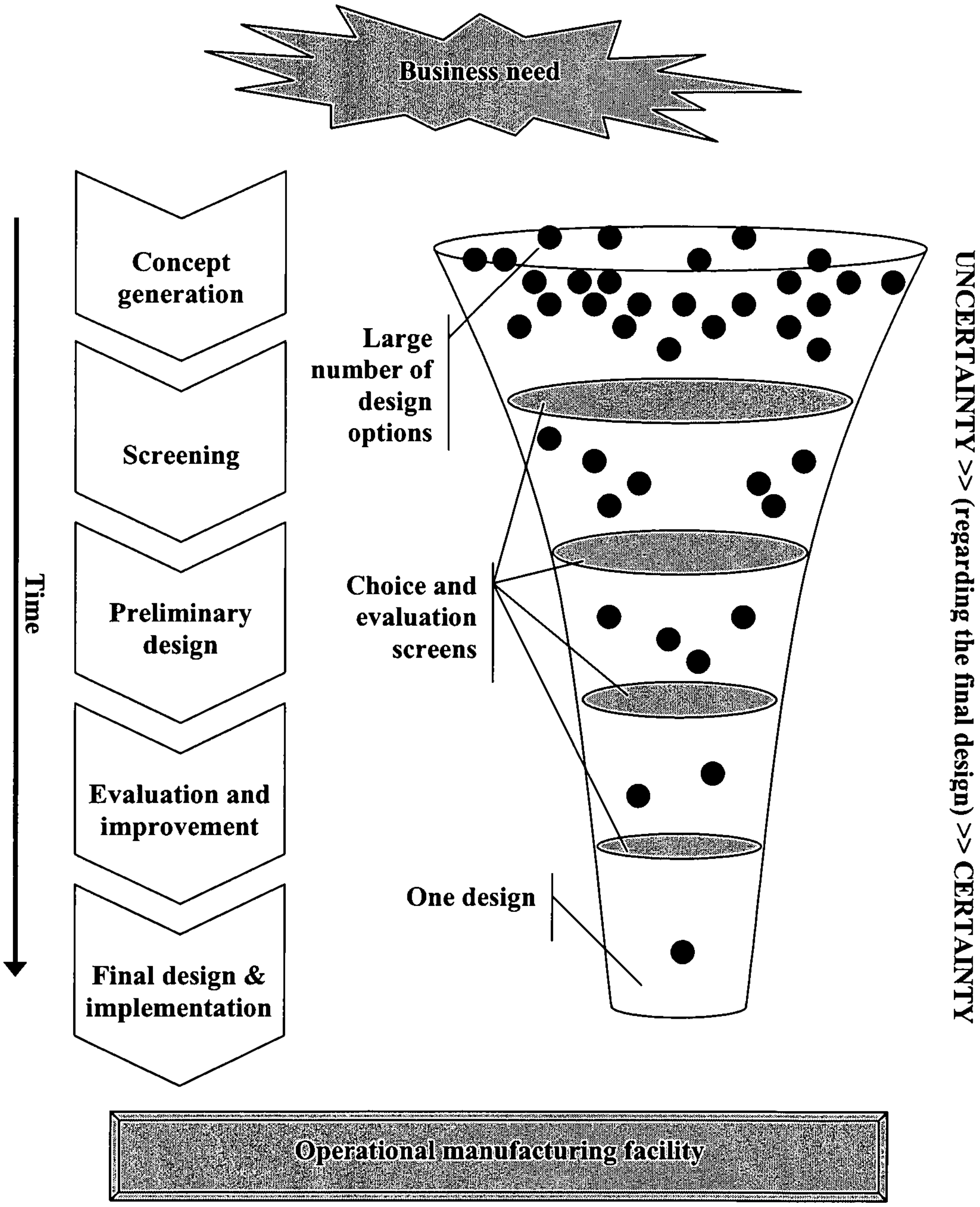


Figure 18: Steps in manufacturing system design and the concept of progressively reducing the number of possibilities until the final design is reached (after Slack et al., 1998).

This process of progressively reducing the number of possibilities until a final design is selected and implemented is crucial because of the relationship between the design phase where the cost is incurred, and its opportunity to reduce the final cost of the product that is being designed. This concept is well documented in the product design and concurrent engineering literature (Baxter, 1995; Krajewski and Ritzman, 1999) and is illustrated in Figure 19. Figure (a), on the left, illustrates that although little is spent (incurred cost) in the early stages of the design process, these decisions determine most of the final cost. Similarly, figure (b), on the right, illustrates how the cost of alteration increases exponentially during the development. Thus, if manufacturing system design can be aided during the early stages with the appropriate tools, modifications in the design can be reduced, and consequently money and time can be saved (Moore, 1999).

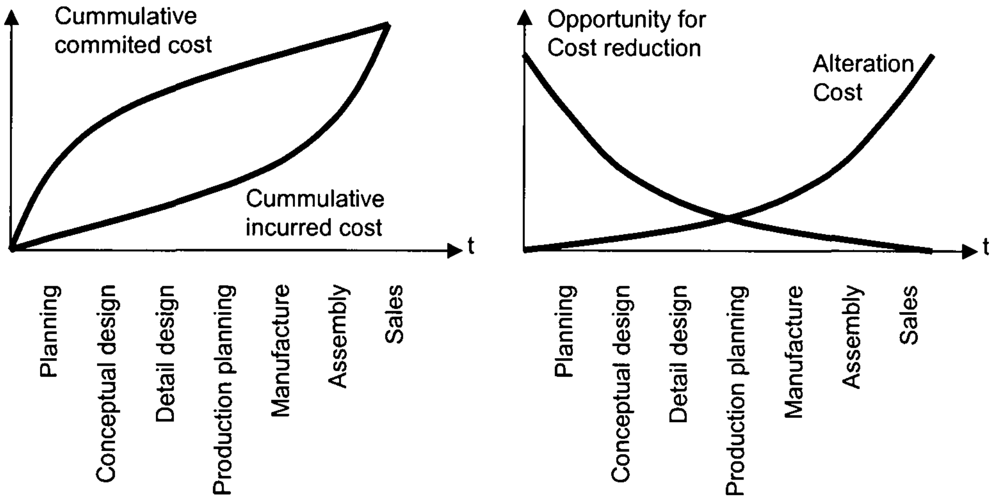


Figure 19: a) Cost committed and incurred in product development and b) Alteration cost and reduction opportunity in product development (Baxter, 1995)

The essence of the design process described above, by progressively reducing the number of feasible possibilities, is that decisions are made in advance of the real product/process being created. This means that the consequences of a particular decision are not always fully known. Thus, all design methods have some form of evaluation, which is usually based on a combination of the analysis, judgement and bargaining of the practitioners involved (Mintzberg *et al.*, 1976; cited in Baines and Kay, 2002). In fact, evaluation is generally considered as a key stage within the design process (Lopez, 2000; Akins, 2002), an opinion that is enhanced by Tognazzini (2000) who not only thinks that a testing plan is the single best indicator of the design process, but also states:

> *"Iterative design, with its repeating cycle of design and testing, is the only validated methodology in existence that will consistently produce successful results. If you don't have testing as an integral part of your design process you are going to throw buckets of money down the drain."*

2.3.3 Importance of simulation within the evaluation stage of the manufacturing system design process

Section 2.3.1 has established the importance of manufacturing to the global economy. It has also highlighted that changes in the competitive environment do occur, and these necessitate changes to manufacturing system designs. Section 2.3.2 outlined the main steps of a typical design process, and showed the importance of the evaluation stage. Thus, the technique used for decision making during the evaluation stage can greatly benefit the design of the manufacturing system, and so improve a company's competitiveness. Simulation is a key technique used by practitioners to evaluate manufacturing system designs (see Section 2.1.4).

Investigations into simulation techniques have persisted for more than 40 years. The 'General Simulation Program' of Tocher in 1958 (Tocher and Owen, 1960 and Tocher, 1979; cited in Page and Nance, 1994) was the first programme developed to apply simulation concepts. Since then, substantial progress has been made in order to transform a 'last resource' method into a valuable technique for aiding industry in many different ways, such as systems analysis, training, etc. As discussed by McKay (2003), *"since the early 1950s, modelling and analytical approaches have clearly dominated how we look at and deal with production issues"*. Slack *et al.* (1998) support this opinion and state that: *"In some ways, simulation is one of the most fundamental approaches to decision making"*. Recent books describe simulation technologies as changing the way in which natural sciences perceive complex systems (Casti, 1997) and the manner in which forward-thinking companies are using simulation to remain competitive (Schrage, 1999). In parallel, industry has found the application of modelling techniques a valuable tool for answering many questions that previous techniques found either too difficult, expensive or even impossible to answer. More specifically, manufacturing systems simulation is generally recognized as a valuable aid to the strategic and tactical decision making in the design process (Carrie, 1988; Robinson, 1994; Banks *et al.*, 1996; Baines and Kay, 2002), especially when used to analyse complex manufacturing systems at both the justification phase and the design phase (O'Kane *et al.*, 2000).

The increasing acceptance of simulation as a valuable tool for industry has been closely related with computer science (Nance and Sargent, 2002). As these two authors state: *"No area within the scope of operations research and the management sciences has been affected more by advances in computing technology than simulation"*. This reason, amongst others, has allowed simulation to remain as a successful technique, enabling it to deal with increasing demands in terms of performance and features. Today's simulation tools enable the user to build computer-based animations of the manufacturing system being considered. While this simulation model is only a coarse replication of the dynamic behaviour of the proposed system, it produces numerical performance indicators, and enables the user to make justified judgements about the real system. However, authors such as Guasch and Piera (2001) state that 'popularisation' of simulation techniques due to the graphical capability of the tools can be a cause of the deficient use of simulation within industry. Literature provides several examples of areas where simulation has been successfully applied in industry. Typical examples of these areas are:

- Supply chain and logistics analysis (Orea, 2000; Sterman, 2000): In this context, simulation can for example be applied to analyse the effect of various shipping policies on cost, delivery lead times and inventory levels.
- Manufacturing (Law and Kelton, 1991; Robinson, 1994): Where simulation can aid the redesign process, through analysing the effects of different policies on factors such as cycle times, costs, inventory levels, identification of bottlenecks, investment, etc.
- Capacity planning (Carrie, 1988; Law and Kelton, 1991): If demand patterns and operation procedures are introduced in the model, simulation can then determine whether planned capacity will be sufficient.
- Contact centres (Roberts, 1978): In this context, simulation can predict, for example, the kind of service level or average waiting time that a company can expect for a particular staffing plan and set of call routing scripts.

Within the scope of manufacturing, simulation can be applied to any system that has entities 'moving' through it, and where data can be obtained on some of the variables of the process that is being analysed. An 'entity' can be defined as *"an object that flows through the simulation model"* (Xu and Abourizk, 1999) and has a clear beginning and an end i.e., it is discrete. Some examples of entities are material through a shop floor, material through conveyor belts, etc. In this context, Askin and Standridge (1993) view the application of simulation superior to simple judgement, because the development of adequate manufacturing simulation models emphasises the process of deciding what the inside of a manufacturing facility should look like (quantitatively and without disturbing or interfering with the real system) and what it should do (study of interactions, taken into consideration even random behaviours). In addition, simulation can be considered to be a predictive technique, rather than just an optimising one. Simulation explores the consequences of decision-making rather than directly advising on the decision itself. This view is shared by Krajewski and Ritzman (1999) who view simulation as an aid to handle more realistic views of a problem and to involve the analyst in the solution process itself.

The scope of simulation in this context is large, and covers, for example, aspects such as the definition of the activities required to convert raw materials into finished products, or the control of the material flows through the facilities. Potter (2000) provides an extensive list of successful applications reported in literature of simulation techniques within the manufacturing context. He classifies these applications into three categories, namely:

1. **Investment decisions**: Analysing capital expenditure, capacity planning, etc.
2. **Design decisions**: Plant layout, Process flow, Line balancing, Human utilisation, etc.
3. **Operating decisions**: Batch sizes, Inventory levels, Training, etc.

Literature also provides examples of advantages and disadvantages obtained by the application of simulation in industry (Fishman, 1973; Shannon, 1975; Vemuri, 1978; Pidd, 1988; Wolstenholme, 1990; Robinson, 1994; Potter, 2000; Sterman, 2000; Klemola and Turunen, 2001; Diaz, 2003). Amongst these authors, it is typically agreed that simulation models allow practitioners to better understand the process, identify its strengths and weaknesses, test options and often develop more efficient alternatives. For example, Pidd (1988) and Robinson (1994) number the advantages of simulation compared with direct experimentation as: (i) Cost, (ii) Time, (iii) Replication, (iv) Safety and (v) Legality. The Simulation Study Group (1991) (cited in Potter, 2000) report identifies *"reduced risk in decision making"* as the simulation benefit most frequently cited in their survey. Simulation techniques also enable practitioners to test a wide range of 'what-if?' scenarios cheaper, easier and quicker, without disturbing the real system (Vemuri, 1978; Sivayoganathan *et al.* 2001). However, some disadvantages of simulation can also be found in literature. For example, the validation process can be difficult when developing simulation models for systems that do not exist in reality (Brinberg and McGrath, 1985). Table 10 provides a brief summary of common advantages and disadvantages of simulation techniques reported in literature.

ADVANTAGES	DISADVANTAGES
* Requires less simplification than mathematical models. * Systems that cannot be accurately described by a mathematical model can be simulated. * Capacity to deal with random effects, present in most processes. * Simulation allows the estimation of performance under some predefined conditions. * Simulation allows the understanding of interactions of potential difficulties. * Simulation often reduces the number of changes in the final system and therefore, manufacturing costs. * Simulation reduces risk, proves concepts and identifies change strategies. * Sensitivity analysis. * Supports / aids the decision. * Reusability and reproducibility. * Time compression / expansion. * Reduce development time; Bring new products to market faster. * Visualization of results. * Operator training.	* Sometimes can be expensive and time consuming develop a correct model. * Because a model is a simplification of reality, the results obtained are not the results of the system, but the results of the model. Thus, a process of validation is required. * If the system does not exist, the validation process can be difficult. * Stochastic models behave differently when random numbers change. * A good technique is required in the simulation process to obtain valid results. * Amount of data required. * Human / technology requirements. * Difficulty of results interpretation. * Inappropriate use of simulation. * Limitations of the tools.

Table 10: Common advantages and disadvantages of simulation techniques.

Although simulation principles are well established, and literature that exhibits a number of advantages of its application within industry exists, its implementation in industry is still not widespread. The Simulation Study Group (1991) found that the level of awareness of simulation within the SME sector was less than 30%, compared with Material Requirements Planning (MRP) at 85% and Computer-Aided Design (CAD) at 80%. This view is supported by a more recent work conducted by Chan and Jiang (1999), who state:

> *"The application of simulation within the manufacturing industry to a greater extent has been limited owing to the fact that simulation has remained within the province of a few. Large companies have employed simulation and reaped the rewards, however, for most medium-sized manufacturers the use of simulation has been, in the past, beyond*

> *their means. The cost of a simulation software package and the technical expertise needed for simulation were the main reasons behind this."*

In summary, the concept of computer simulation as a legitimate tool in the design and analysis of new and existing manufacturing systems has been well documented (Mills, 1993; Hollocks, 1995; Chan, 1995; Chan and Jiang, 1999; Guasch and Piera, 2001). However, evidence based upon the real and quantitative benefits of simulation are actually very rare, being mostly found with marketing purposes in software vendors web pages. Such data may be available within companies, but due to the sensitivity of such information, has not reached the public domain. There are some isolated examples of how independent practitioners have quantified the benefit of simulation, such as Gallaher and Martin, (1999) and Guasch and Piera (2001) who state that *"a recent study sponsored by the European Union quantify the impact of simulation in industry as an increment of 5 to 10% in the productivity"*. However, no macro analysis of benefits exists. Therefore, an empirical agreement for simulating a manufacturing system can, at the best, only be considered to be weak.

2.4 Current issues that constrain the application of simulation in industry

The final section of the literature review includes an analysis of the characteristics that constrain the wide spread of simulation in industry. This is done by analysing the process of a typical simulation study, in order to identify the tasks that require more effort (Section 2.4.1). Then, Section 2.4.2 focuses on the particularities of model development, including acquisition of data and model building. Finally, Section 2.4.3 describes the main characteristics of modern simulation tools.

2.4.1 Characteristics of a simulation study

Section 2.3.3 has shown the importance of simulation within the evaluation stage of the manufacturing system design process, focusing on the advantages and disadvantages of the simulation technique within this task. However, it is necessary to emphasise that the creation of a model and its coding are just part of an overall simulation effort to understand or design a complex system (Law and Kelton, 1991). Thus, this section offers an overview of the main stages involved in a simulation project and the relationships among them.

In practice, although the important role that simulation can play in analysing production systems has now been generally realised, its use is not necessarily straightforward (Centeno and Carrillo, 2001). Simulation techniques have often been associated with 'art' rather than 'science' (Vemuri, 1978). However, the adoption of a systematic methodology to use with simulation techniques brings simulation nearer to the scientific area and benefits its use, amongst other reasons, by facilitating it and increasing its repetitivity. This view is supported by McHaney (1991), who matches the different steps of a scientific method with the corresponding simulation activity (see Table 11).

STEP IN SCIENTIFIC METHOD	CORRESPONDING SIMULATION ACTIVITY
Problem definition.	Setting simulation objectives.
Formulating hypothesis.	Defining model scope and detail. Selecting modelling view, language and coding model.
Experimentation.	Running model.
Results.	Obtain data from model.
Conclusion.	Using statistics and judgement to evaluate.

Table 11: Scientific methods in simulation (McHaney, 1991)

The process of application of simulation techniques has been detailed by authors such as: Shannon, (1975); Carrie, (1988); Law and Kelton, (1991); McHaney, (1991); Robinson, (1994); Kheir, (1995); Banks *et al.*, (1996); Oakshott, (1997) and Guasch and Piera, (2001). These authors view the application of simulation mainly as a sequential process, but with iterations within the different steps. The number of steps described by the above mentioned authors vary substantially (while Carrie (1998) mentions 6 stages, McHaney (1991) and Robinson (1994) describe 15), but mainly due to the decomposition of a given stage into more detailed ones and vice versa, rather than different views in regard to the simulation process. These steps are shown in Table 12 and typical simulation project stages are graphically depicted in Figure 20. Appendix B expands this concept and offers an overview about the different activities involved in each stage of a simulation project.

STAGES / AUTHORS	1	2	3	4	5	6	7	8	9
1.- Shannon, (1975); 2.- Carrie, (1988); 3.- Law and Kelton, (1991); 4.- McHaney, (1991); 5.- Robinson, (1994); 6.- Kheir, (1995); 7.- Banks et al., (1996); 8.- Oakshott, (1997); 9.- Guasch and Piera, (2001)									
1.- Problem formulation, objectives and plan	●	●	●	$\odot_4$	$\odot_3$	●	$\odot_2$	●	●
2.- Model conceptualisation	●		●	●	●	N	●	●	●
3.- Data collection	●	●		●	●	●	●	●	●
4.- Model translation / building	●	●	●	$\odot_3$	●	●	●	●	●
5.- Model verification	N	N		●	N	N	●	●	●
6.- Model validation	●	●	$\odot_2$	●	●	●	●	$\odot_2$	
7.- Experimental design and runs	$\odot_3$	●	$\odot_2$	●	$\odot_2$	●	●	●	●
8.- Analysis of experiments	●	●	●	●	●		●	●	●
9.- Documentation and reporting	●	N	●	●	$\odot_2$	●	●	$\odot_2$	●
10.- Implementation and/or training	N	N		●	$\odot_3$	N	●	N	●
TOTAL NUMBER OF STAGES	**10**	**6**	**9**	**15**	**15**	**6**	**11**	**11**	**9**
Note: ● = Considered; $\odot_N$ = Decomposed in 'N' sub-stages; N = Not considered explicitly.									

Table 12: Steps in a simulation study

The steps defined in Table 12 were linked to form a flow diagram that reflects the interactions between the different stages of the project. In addition, the diagram has been complemented by indicating the division of responsibilities of each stage after the recommendations of McHaney (1991). However, neither Table 12 nor Figure 20 show the time involved in each stage. As stated by Shannon (1975), *"the probable cost and time of the simulation should always be weighed against the value of the information it is likely to produce"*. This concept is relevant, because although simulation is often the preferred technique to solve complex manufacturing systems design problems in industry (see Section 2.3), limitations on its application (due to expensive data collection processes, construction of complex models, skills, etc.) can constrain its expansion in industry, and benefit alternative techniques such as mathematical ones (see Section 2.1.3).

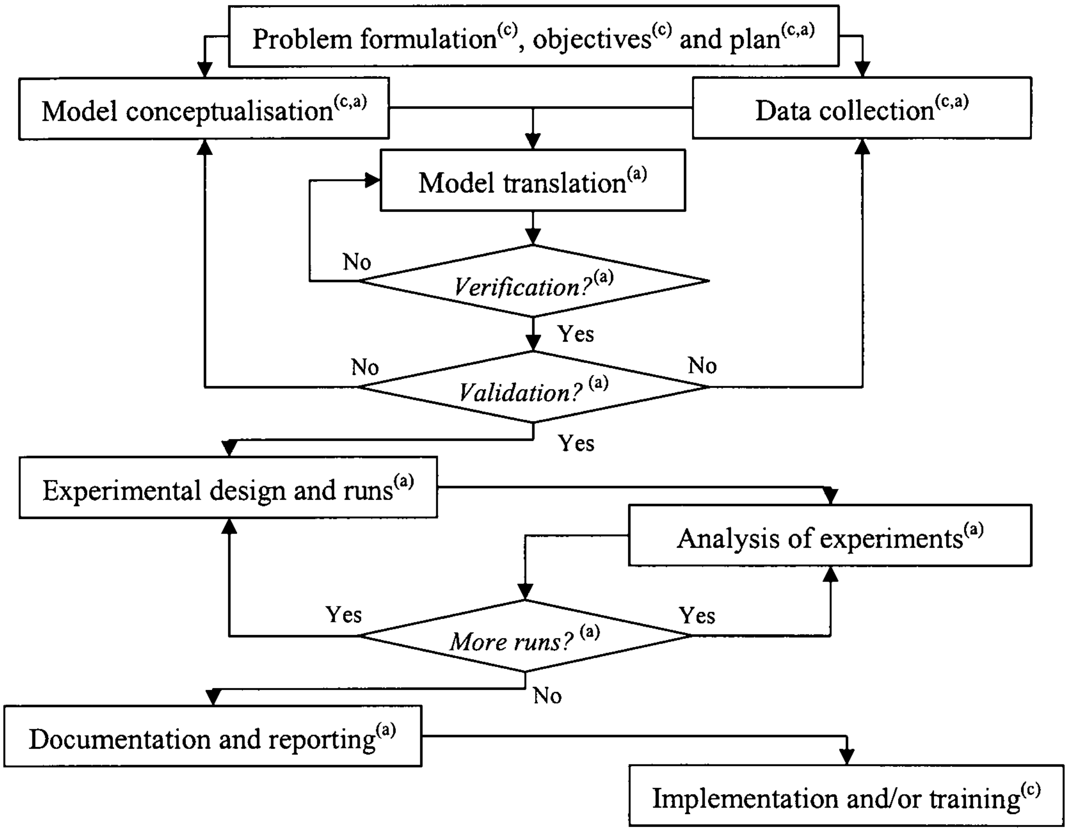


Figure 20: Steps in a simulation study (after Shannon, 1975) and division of responsibilities: (c) Simulation Customer and (a) Simulation Analyst (after McHaney, 1991)

Robinson (1994) provides an estimation of the time required in each stage of the simulation project, grouped in four main areas: (i) Problem definition, (ii) Model building and testing, (iii) Experimentation and (iv) Project completion. Similarly, based on a number of industrial applications, Trybula (1994; cited in Liyanage and Perera, 2000) suggests that in a typical simulation project, 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%
- Conclusions and recommendations ≈5%

This view is supported by Gershefski (1970) (cited in Shannon (1975) who believes that data collection and model development take 25% and 40% of the total project time, respectively. Obviously the actual proportions vary greatly depending on the specific project, especially for experimentation (Robinson, 1994). Typical factors that have an effect on the total time spent are: (i) Model size, (ii) Model complexity and (iii) Time to experiment. In addition, Robinson (1994) agrees that model building (including the model development and data gathering) takes a significant proportion of the time. This problem is usually emphasized because the model building task is often made more complex than necessary (Banks *et al.* 1996), due to the tendency *"to simulate too much detail rather than too little"* (Shannon, 1975).

As can be appreciated from the data provided by Trybula (1994; cited in Liyanage and Perera, 2000), time required for model experimentation is usually less than time required for model development or data gathering. Reasons for this are the increased computer performance and the practice of parallel or distributed simulation (Swain, 1999), where the computer load is shared by a number of computers within a network. However, as Henriksen (1998; cited in Banks, 1998) states, advantages provided by distributed simulation can sometimes be considered as backward steps, because this practice can encourage the user to build more detailed models. Authors such as Law and Kelton (1991); Petropoulakis and Giacomini (1998); Sawhney *et al.* (1999); Rabbath *et* al. (2000); Sarjoughian and Zeigler (2000); Schaefer and Wolfe (2000); Holst and Bolmsjo (2001); Zobel (2001) and Nikolaidou and Anagnostopoulos (2003) provide more detailed information about the characteristics involved with parallel simulation.

2.4.2 Characteristics of a model development

The previous section has shown a typical distribution of time within the different tasks involved in a simulation project and highlighted the importance of the data collection and modelling stages. It has also stated that there is a tendency to simulate too much rather than too little. This common practice leads into an increasing demand of data to include in the model, making this stage (data collection) a major task (Banks *et al.*, 1996). When modelling stochastic systems, a decision must also be made whether to use empirical data directly in the model or to use theoretical probability or frequency distributions (Shannon, 1975). Empirical data are useful when the system does not change substantially, otherwise the model will reflect the past and not the desired period of time. On the other hand, distributions are processed by computers in a more efficient manner, accelerating the running time. In any case, decisions regarding the amount and type data to be used and their validity are all critical to the success of the simulation project (Fishman, 1973; Shannon, 1975).

Thus, several alternatives can be followed to reduce the amount of time required to accomplish the model development stage (including data collection), and therefore, reduce the final time, or increase the time dedicated for experimentation (Robinson, 1994; Perera and Liyanage, 2000). These are: (i) Reduce the complexity of the model, (ii) Analyse a suitable simulation methodology and (iii) Integrate the modelling tool with existing databases. The election of one (or various) of these alternatives depend on the characteristics of the project in hand and the resources and skills available.

2.4.2.1 Model's complexity

According to Robinson (1994), when considering what to include in a model the basic rule is:

> *"Model the minimum amount of detail required to achieve the project's objectives"*

The reasons for this are illustrated in Figure 21. This figure represents the expected accuracy of a simulation model in relation to its scope and level. While Figure 21a represents the expected accuracy of a simulation model in relation to its scope and level, Figure 21b shows the time required to build a model. Both figures are divided into three zones, as follow:

- **Zone I:** It is represented in the left area of each graph. As can be seen in the figure, increasing the scope and level leads to significant gains in accuracy. As stated by Robinson (1994): *"basically, 80 per cent of the accuracy is obtained from 20 per cent of the model detail"*
- **Zone II:** In this zone, the advantage of further increases is not as great.
- **Zone III:** Once too much detail is added, it might be difficult to find quality data for new attributes and the accuracy of the model might actually be reduced (Robinson, 1994; Perera and Liyanage, 2000); hence the dip in the curve.

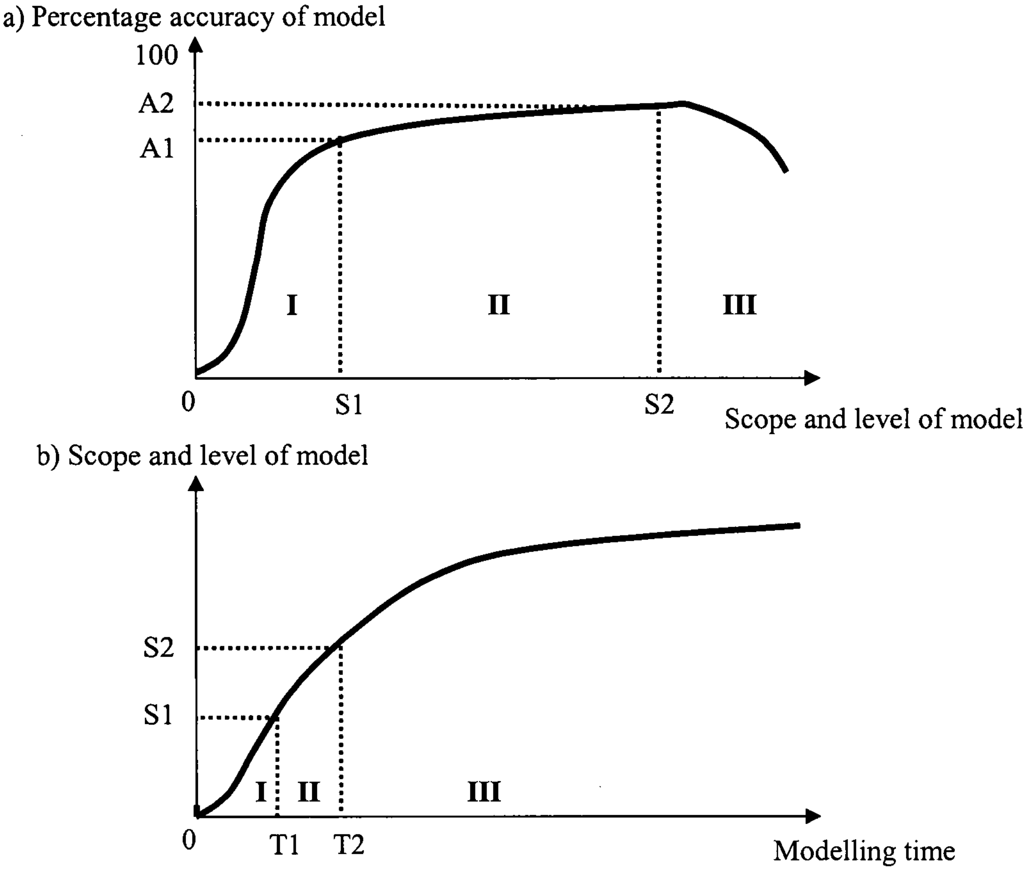


Figure 21: Model accuracy, scope and level, and modelling time (Robinson, 1994)

Perera and Liyanage (2000) provide further information about the effort required to collect the different categories of data. As a summary, practitioners that participate in the survey required the most effort when collecting 'breakdown data' and the least effort when collecting 'process times' or 'process routes'.

2.4.2.2 Simulation methodologies

The simulation methodology adopted within a project also has an effect on the time spent during the different phases of the project and the data requirements (Baines, 1994; Lin *et al.,* 1997; Oyarbide *et al.* 2003). An example of this can be found between the DES and SD techniques; the manner in which these techniques treat the elements (discrete vs. continuous) or the way in which these techniques are used (SD is usually used in a more aggregated level of detail) (Lin *et al.,* 1997). The main particularities of these techniques, identified by Carrie (1988), are provided in Table 13. In addition, Pidd

(1988) details the advantages and disadvantages of both 'time-slicing' and 'next-event' methods and the scenarios in which they might be better suited.

	Type of model		
	Continuous (theoretical)	*Continuous (digital)*	*Discrete*
Time step	Infinitesimal	Small time slices	Jumps from one event to the next
Method	Differential equations	Difference equations	Logical relationships
Components	Aggregate	Aggregate	Individual entities
Variables	Levels	Levels	Queues, states, attributes
Changes	Rates	Rates	Events

Table 13: Characteristics of different types of modelling (Carrie, 1988)

Research conducted by Baines (1994) and Baines *et al.* (1998), comparing the model building time and accuracy relationships of common modelling techniques (IDEF$_0$, IEM, DES, SD, QT, ABC and BP; described in Section 2.2), revealed that simulation techniques (SD and DES) were substantially more accurate than other forms of modelling. It also revealed that the SD technique *"has slightly less flexibility* (than DES) *but it exhibits a relatively rapid model build rate and model execution time".* However, this study also found that DES models can be more accurate and credible than models built using the SD technique if time and data are not a limitation. However, a study conducted later by Barton and Tobias (1998) revealed that errors produced because of the method of calculation over discrete time intervals in SD models can be significantly reduced.

As stated in Section 2.1.4.3, hybrid simulation techniques do exist (Petropoulakis and Giacomini, 1998; Donzelli and Iazeolla, 2001; Martin and Raffo, 2001; Lee et al, 2002; Kiesmuller, 2003). Since some systems are neither completely discrete nor completely continuous, hybrid models can be constructed in order to model the real system in a more efficient manner (Van Beek and Rooda, 1997). However, Coyle (1985) and Lane (2000) recognise that simulation modelling (and tools related with it) has traditionally been divided into two separate types (DES and SD), but also state that systems where both continuous and discrete processes exist can be solved without the use of hybrid languages, by representing discrete events in SD models. This view is supported by vendors of simulation software, who are gradually integrating both continuous and discrete elements within their products (Pidd, 1999).

2.4.2.3 Integration of simulation tools

Data interface problems are a factor that inhibit the use of manufacturing simulation (McLean and Leong, 2001). The data collection stage does not comprise only the acquisition of data (from sources such as computer print-outs to shop floor handwritten data), but also its conversion into an adequate format, and many times, the re-typing into the model (Mason *et al.* 1998). Thus, project time could be significantly reduced if simulation tools had the ability to pull data directly from any standard MRP/ERP system or any database or spreadsheet product. This would eliminate a lot of project time (and errors) used for data collection, and allow the practitioners to focus more time on model logic and validation (value-added activities) (Buxton, 2000; cited in Banks, 2000b). In addition, this would allow the model to be easily updated by the client in the future if the data are changed.

2.4.3 Characteristics of the computer tools

Computer tools were not popular in the origins of simulation techniques (DES and SD). However, as stated in Section 2.3.3, the popularisation of simulation techniques within industry has been closely linked with the development of more powerful computers and the design of user friendly computer tools with extensive capabilities. This evolution has been especially notorious since the mid 1980s, mainly due to the widespread availability of personal computers (such as IBM PC compatibles and Macintosh) with graphical interfaces (Pidd, 1988). Nowadays, hundreds of graphical tools exist (see Directory of Simulation Software (DSS); edited by The Society for Computer Simulation), allowing a wider range of users to apply simulation due to their user friendly interfaces (Nance, 1995; Rauniar *et al.,* 2002). This concept has been explained in detail by authors such as Hlupic and Paul (1996); Hlupic (1999a; 1999b, 2000) and Hlupic *et al.* (1999) who analyse the main characteristics and evolution of simulation software, or Valentin (2001), who develops an extensive simulation software comparison. For further information, refer to Altmann (1996).

Simulation tools based in both DES and SD methodologies have evolved differently to meet the requirements of their users and applications. However, software developers have considered it important to enhance the graphical capabilities and operability of their tools, indistinctive of the methodology chosen. Thus, SD based tool have evolved from tools such as DYNAMO or DYSMAP, where users were required to write the model in terms of equations, to tools such as iThink/Stella, a visual interactive modelling system (VIMS) based on graphical user interfaces (GUIs) that can construct models graphically with the aid of a mouse. However, other features have evolved very differently amongst DES and SD tools. While DES based tools such as Witness, ProModel, etc. tend to provide a set of components tailored to behave in a predetermined manner, SD tools still construct models in a similar manner (levels, flows and rates) to those the originally developed by Forrester (Pidd, 1988). According to Pidd (1988), SD diagrams tend to get very complicated once a system reaches a certain scale.

Findings obtained from a simulation software survey conducted by Hlupic (2000) amongst academic and industrial users is summarised below:

1. Most applications of simulation are in the area of manufacturing systems. (83% of academia users; 33.3% industrialists).
2. A typical number of simulation packages used is one or two (55.4%).
3. In academia, most popular DES based tools are Simul8 (44.4%) and Witness (38.8%); SD based tool is Stella/iThink (<5%). In industry, Witness (22.2%) is the most popular DES based tool. There are no SD based tools amongst the top nine.
4. 44.4% of academia believes that tools have a lack of modelling facilities. However, 38.8% of them believe the opposite. In industry, 33.3% of respondents believe that simulation is a powerful tool, but also that flexibility is reduced (22.2%).
5. 22.2% of academia thinks that tools are difficult to learn; in industry this percentage is 11.1%. However, industrialists are more concerned about the price of the tools (22.2%) and limitations regarding data input (22.2%).
6. The most appreciated features within simulation tools are: Easy to modelling (61.1% of academia; 22.2% of industrialists), Visual facilities / animation (50% of academia; 33.3% of industrialist) and Flexibility to link to external code (22.2% of academia).

The above findings are consistent with the literature. For example, ‘Finding 1’ shows that simulation is commonly used in the area of manufacturing, and thus, shows the acceptance of simulation as a valuable tool for this task. In addition, it is apparent that an appropriate selection criteria for choosing the right simulation tool is crucial, due to the incurred purchasing cost (from $25 to > $20,000; DSS), training and incompatibility with other simulation tools amongst other reasons (see 'Finding 2' and 'Finding 5') (Oakshott, 1997; Chan and Jian, 1999).

In practice, DES based tools are the most used ('Finding 3'). This view is supported by Baines and Harrison (1999) who, after analysing 80 papers within the simulation context, stated that *"manufacturing applications have been less widespread"* within the SD technique. However, in the view of Baines and Harrison (1999), *"research into the application of SD to the modelling of manufacturing systems is encouraged. In particular, a dedicated software tool for manufacturing system modelling is lacking"*. For specific information about simulation tools, refer to the 'Proceedings of the 1996 Winter Simulation Conference', which includes almost 30 papers where software vendors present information about their products.

Finally, 2D and 3D visual interfaces with graphical animation and easy to use interfaces raise the value of simulation tools (Finding 6) (Au and Paul, 1996). Potter (2000) describes several benefits of the inclusion of 2D and 3D animation capabilities within computer tools; this view is also supported by many authors in this field (Law and Kelton, 1991; Robinson, 1994; Sterman, 2000; etc.). However, graphical notation seems to be dependent on the technique used (see Section 2.2), making the shift between techniques more difficult.

2.5 Summary

This literature review has focused on the concepts of simulation and the characteristics of their most popular techniques (DES and SD), emphasising the importance of the manufacturing role in industry and the relevance of the simulation techniques within this context. The review began in a wider sense by considering the nature of modelling including definitions. Following this, the concepts of modelling were explained and a classification of model types (including overviews of the mentioned techniques) was provided, identifying both DES and SD as symbolic analytical techniques. Characteristics of these simulation techniques (DES and SD) were then detailed and the diagramming conventions associated with them where also described. While original diagramming conventions of both DES and SD can be beneficial in some contexts, they appear to look complicated when modelling complex systems. This concept seems to be accepted by DES tool developers (creating multifunctional standard elements); however, SD based tools still keep the original 'level and rates' notation. The literature continued by reviewing the importance of manufacturing and the process followed to its design; followed by a discussion about the role of simulation in the evaluation stage of the process mentioned previously (showing the popularity of DES in respect to SD), and the benefits of including simulation earlier in this process. Finally, this literature review describing the main issues that constrain the application of simulation in the manufacturing system design process; these issues were classified in terms of: (i) Process, (ii) Model development and (iii) Computer tools. DES was found to be the main form of simulation in industry. Problems associated with its application, such as 'over simulation' and excessive data requirements were identified and the non linearity between the time spent and results accuracy was identified. Ineffectiveness of other modelling techniques such as Queuing Theory were highlighted, and the need for more sophisticated modelling tools, such as simulation, was revealed. Within simulation, DES was found to be the prevalent tool, but literature also revealed that DES is not used in a very efficient way in industry. In addition, benefits of the application of continuous simulation methods such as SD were listed, and the suitability of the SD technique as an aid to the manufacturing systems design was discussed. Thus, this literature review revealed that the investigation of continuous simulation based on the SD technique can be a valuable area of research. The need for simulating the systems quicker through a simpler model, a more efficient technique, or simply by using a faster computer is mostly motivated by a desire to gain competitiveness. Thus, this concept will be taken further in the following chapter, by describing the research problem, aim and objectives.

Chapter 3

RESEARCH AIM AND PROGRAMME

Chapter 2 introduced the concepts of modelling and simulation, and highlighted the importance of simulation within the manufacturing industry. This chapter builds upon these foundations to develop the research aim and programme. It is structured in five sections, as follows:

1. Problem definition.
2. Development of research aim and objectives.
3. Research scope.
4. Principles of research design.
5. Formation of research programme and methods.

3.1 Problem definition

Classical modelling techniques, such as the 'Queuing Theory', have been used in the past with considerable success to tackle a wide variety of simple problems (Allen, 1990). However, their ineffectiveness in dealing with complex real-life manufacturing design issues has led to the growth of more sophisticated modelling techniques, such as simulation (Section 2.3.3). Indeed, manufacturing systems have reached a level of complexity unimaginable some years ago, and it is generally agreed that modelling, and more explicitly simulation, is a useful aid when attempting to design or redesign such systems (Section 2.3.3).

To deal with such level of complexity, one of the solutions adopted by industry has been to exploit computer based simulators based on Discrete Event Simulation (DES) technique. The power and graphical capabilities of DES-based tools have replaced other ways of simulation when a considerable level of detail is required in complex systems (Section 2.2). However, DES has been criticised because the resultant models tend to be complex, time consuming and requiring large amounts of data (Aitchison, 1995) (see Section 2.4).

In a similar way, simulation based on the System Dynamics (SD) technique has achieved good results when attempting to understand the behaviour of complex systems in different fields (Sastry, 1993). SD was originally designed for the modelling of industrial systems (Forrester, 1969); this technique is highly credible, and has some characteristics that can make it desirable for simulating some manufacturing scenarios (Section 2.2 and Section 2.4). Unlike DES, for example, the SD technique is particularly capable of modelling complex systems quickly (Sterman, 2000). However, SD is rarely applied to evaluate manufacturing issues, whereas DES has been widely adopted.

Evidence suggests that the performance of computers has an influence on the utilisation of computer based tools when applying simulation techniques (Section 2.3.3 and Section 2.2). It also seems clear that the success in the adoption of a modelling technique is dependant not only on the technique itself, but in the software tool that allows its application. The significant improvements produced in this area in the last few decades have allowed companies to produce more simulators that are able to deal with a wider variety of scenarios, include more features, and obtain results faster. For example, currently available DES tools benefit users from improved user interfaces and customisation of the tools. Although this should lead into 'easier' simulators, the compromise between including 'more features' and making the software 'easy to use' means that, in practice, most known tools available on the market are useful only for highly skilled users.

SD based tools have also benefited from improved user interfaces and the inclusion of more sophisticated features. However, SD tools are currently configured as generic building blocks, and this can mean that such tools are difficult to use (Lin *et al*, 1997). This might well explain why SD has much less penetration than DES to manufacturing system design (Baines and Harrison, 1999), and hence an opportunity might be being missed.

These arguments lead to the research question, which can be summarised as follows:

"Can the modelling of manufacturing systems be improved by incorporating the principles of SD within a modelling tool that is tailored to this task?"

3.2 Development of research aim and objectives

The research problem outlined above led to the development of the following research aim:

To determine the extent to which System Dynamics can provide a credible alternative to Discrete Event Simulation in the process of manufacturing system design.

There are a number of research issues that naturally need to be addressed for the fulfilment of this aim. Therefore, the main objectives of this research have been developed. These are as follows:

1. Understand what is needed of a SD modelling tool for it to complement the needs of a manufacturing system designer.
2. Represent the capabilities of SD in a modelling tool tailored to manufacturing system design.
3. Assess the true capabilities of SD, by applying the modelling tool to real manufacturing problems, and assessing performance against a typical DES modelling capability.

The originality of this work will arise from the knowledge gained about the real capability of SD to assist with manufacturing problems. This will have an impact on manufacturing competitiveness because it will enable companies to make better decisions about the development of their manufacturing business.

3.3 Research scope

Based on the objectives mentioned above and the literature review, the scope of this research can be summarised as follows.

- ➢ **Domain:** This research focuses only on manufacturing systems modelling due to its relevance in industry (Section 2.3.1).
- ➢ **Simulation Techniques:** The research is restricted to the DES and SD graphical computer-based simulation techniques due to their large range of applications and suitability to deal with complex problems (Section 2.1.4, Section 2.3.3 and Section 2.2).

- **Development of a tool based on the SD principles:** The tool to be developed as part of this research will focus on the core features analysed by industry when simulating the manufacturing processes, such as productivity, throughput time, work in progress, etc. (Section 4.3)
- **Performance Analysis:** The performance of simulation techniques will be analysed using industrial case studies that satisfy the objectives of this research (Chapter 6).

3.4 Principles of research design

In order to achieve the aim and objectives outlined above, a research strategy or methodology has been developed. This section will examine the different methodologies and concepts available. Following this, the actual methodology adopted by the researcher will be described.

Research methodology refers to the theoretical analysis of the methods appropriate to a field of study, or to the body of methods and principles particular to a branch of knowledge. In recent years however, 'methodology' has been increasingly used as a pretentious substitute for 'method' in scientific and technical contexts (Bready, 2000).

Merriam-Webster's Unabridged Dictionary (2000) defines the word 'method' as:

> *"A means or manner of procedure, especially a regular and systematic way of accomplishing something"* or
>
> *"Orderly arrangement of parts or steps to accomplish an end"*

While 'methodology' is defined as:

> *"A body of practices, procedures, and rules used by those who work in a discipline or engage in an inquiry; a set of working methods"*

Thus, 'methodology' can be perceived as a logical framework that is designed to enable the achievement of the research aim within its constraints, such as timescale or resources. This argument is reinforced by Phillips and Pugh (2000), who states:

> *"Methodology helps to ensure project aims are achieved and facilitates the process of answering of the research questions and meeting the deliverables"*

For the reasons described above, for a given research aim, there is no one single methodology that should be followed, because it will strongly depend of the research constraints. Moreover, it is reasonable to state that there is no perfect methodology (Garson, 2002), although the strategy and tactics selected in carrying out a piece of research will be guided by the type of research question that needs to be answered (Manstead and Semin, 1998; cited in Robson, 1993).

A number of books do exist to help researchers select an appropriate methodology when conducting research. Examples are Alreck and Settle (1985), Burns (2000), Gill and Johnson (1991), Greenfield (1996), Robson (1993; 2002) and Yin (1994). These books tend to focus on some areas of research methodology (quantitative, qualitative, etc.) or on some areas of application (academia, social sciences, etc.). An exception to this is Robson (1993; 2002) who is the most generic of all the above authors. The research process is presented as a clear and logical structure, with consideration of most if not all appropriate research methods. Therefore, in agreement with previous Cranfield University researchers in this area (Ince, 2000; Ford, 2001; Adesola, 2003; Bhamra, 2003) basic guidelines for defining and designing an appropriate research methodology are observed here.

According to Robson (1993), a correct methodology design needs to take the following aspects into consideration:

1. Identification of the research purpose.
2. Selection of the research strategy and research type.
3. Data collection methods.
4. Analysis of data and evaluation.

3.4.1 Research purpose

The first step in undertaking a research project is to define the purpose of the research. Robson (1993) classifies the purposes of enquiry in three groups; namely: (i) Exploratory, (ii) Descriptive and (iii) Explanatory (see Table 14). Exploratory research is concerned with finding out what is happening and often involves using case studies and/or surveys. Descriptive research, as its name suggests, aims to provide an accurate profile of an established situation, surveys being an adequate technique for gathering data. Finally, explanatory research seeks an explanation of a situation or problem. Experimentation is an accepted method for data gathering and analysis for this purpose.

Type	Characteristics
Exploratory	To find out what is happening. To seek new insights. To ask questions. To assess phenomena in a new light. Usually, but not necessarily, qualitative.
Descriptive	To portray an accurate profile of persons, events or situations. Requires extensive previous knowledge of the situation etc, to be researched or described, so that you know appropriate aspects on which to gather information. May be qualitative and/or quantitative.
Explanatory	Seeks an explanation of a situation or problem, usually in the form of causal relationships. May be qualitative and/or quantitative.

Table 14: Classification of the purposes of enquiry (Robson, 1993)

The purpose of the research is essentially dependent on the research aim. The research aim in this thesis is concerned with investigating the SD technique within the design of manufacturing systems. In other words, to explore if the SD technique is suitable for modelling manufacturing systems or there are justified reasons to discard the SD technique within this task. Thus, the purpose of this research can be considered as exploratory in nature.

3.4.2 Research strategy and type

Research strategy is conditioned by the research purpose. Robson (1993) categorises the research purposes into three main groups, as follows:

1. Experiment, measuring the effects of manipulating one variable on another variable.
2. Survey, collection of information in standardised form from groups of people.
3. Case study, development of detailed, intensive knowledge about a single 'case', or of a small number of related 'cases'.

For those research purposes that can not be achieved by the use of one of the strategies described above, Robson (1993) also notes that there are other forms of strategies that can be employed, such as a 'hybrid strategy' (one that falls between the three basic types of strategy), being in some particular cases the only suitable strategy to fulfil the research needs.

The application of these research strategies depends on three main factors (Robson, 1993 and Yin, 1994): (i) Research question/s; (ii) The degree of control over events; (iii) Focus on past or present events. A detailed explanation of the three main types of research strategies and their areas of application can be seen in Table 15.

In a similar way, the research purpose indicates the 'type' of research conducted, usually in the form of 'qualitative' and/or 'quantitative' research. Saunders *et al.* (1997; cited in Sherwin, 2000) provide some typical features of these two types of research, summarised below:

Qualitative research:

- Based on meanings expressed through words.
- Results in non-standardised data requiring categorisation into categories.
- Analysis conducted through the use of conceptualisation.

Quantitative research:

- Based on meaning derived from numbers.
- Collection results in numerical and standardised data.
- Analysis conducted through the use of diagrams and statistics.

This concept has now been taken forward and used in the formation of a research strategy in next section.

Table 15: Summary of the three basic research strategies (after Robson, 1993)

Strategy	CHARACTERISTICS		APPLICATION		
	Description	Typical features	Type of research question	Requires control over events?	Focus on current events?
Experiment	Measuring the effects of manipulating one variable on another variable.	Selection of samples of individuals from known populations Allocation of samples to different experimental conditions. Introduction of planned change to one or more variables. Measurement of a small number of variables. Control of other variables. Usually involves hypothesis testing.	How, Why.	Yes	Yes
Survey	Collection of information about standardised information from groups of people.	Selection of samples of individuals from known populations. Collection of a relatively small amount of data in standardised form from each individual. Usually employs questionnaire or structured interview.	Who, What, Where, How many, How much.	No	Yes
Case study	Development of detailed, intensive knowledge about a single 'case', or a small number of related 'cases'.	Selection of a single case (or a small number of related cases) of a situation, individual or group of interest or concern. Study of the case in its context. Collection of information via a range of data collection techniques including observation, interview and documentary analysis.	How, Why.	No	Usually but not necessarily

3.4.3 Data collection methods

Gill and Johnson (1991) suggest that the main concern of any research method is how to tackle tasks. Robson (1993) says that it is not necessarily good research practice to conduct an investigation by using only one method. Similarly, Denzin (1988) states that the use of different methods, sources and investigators achieves the triangulation that is important for increasing the credibility of a study. This is known as a 'multi-method' approach. Table 16 shows the most commonly employed data collection methods.

Questionnaire	Using 'yes/no' type answers. Good data for quantitative type research. Usually part of a survey.
Descriptive questionnaire	Descriptive answers more suited to qualitative type research. Usually part of a survey.
Interviews	Can be structured or semi-structured in format.
Observation	Used to report what people do and not what they say they do.
Ethnography	'Going native' type of observation where the researcher is immersed in the environment.
Documents	A good source of historical information.
Workshops	Used to discuss specific issues with a number of people simultaneously.

Table 16: A selection of data collection methods

Collins and Cordon (1997) suggest that face-to-face interviews are preferred to telephone interviews because they avoid 'adulteration'. They divide these interviews in three categories:

1. **Structured**: While it has the advantage of repeatability, rigidity and lack of flexibility can be a constraint.
2. **Semi-structured**: It allows a discussion beyond the pre-established questions, alteration of questions order or even the inclusion of new questions or removal of the existent ones.
3. **Unstructured**: This category enables free discussion of facts and opinions.

These ideas are followed and expanded within the first stage of this research (Section 3.5.2) for selecting a suitable data collection method for understanding business needs.

3.4.4 Analysis of data

Robson (1993) builds a table of rules for analysing qualitative data, which are based upon the work of earlier authors that include Miles and Huberman (1984) and Delamont (1992). Robson (1993) advances the notion that there is no 'right' way of analysing this kind of data, although the researcher should be systematic and organised. Another rule states that themes, categories and codes should be generated as the researcher proceeds. Importantly, he states that the main tool in analysing qualitative data are comparison. Software based qualitative analysis tools such as QualPro and NUD-IST (non-numerical data indexing searching and theorising) for example are well known in the sphere of social science research, although they are best suited when there are large amounts of data to process.

The reliability of data can be further increased with the use of multiple sources of data (triangulation), which is an important notion in case research (Voss *et al*, 2002). An example of triangulation that appears particularly relevant to this study is by Boyer and McDermott (1999), and cited in Voss *et al* (2002):

> *"To augment the on-site interviews and surveys, tours of the manufacturing facility were arranged. These tours allowed for a visual check and comparison of each firm's efforts in such areas as AMT adoption, layout, degree of worker empowerment and training, and technology relative to others in the industry. In general, these plant tours provided an opportunity for verification and clarification of survey and interview responses, as well as providing the researchers with a feel for the overall work environment and systems."*

3.5 Formation of research programme and methods

To realise the above aim and objectives within the stated scope, a strategic research programme is necessary to direct the activities of this research in a sequence of stages. These stages have been combined with a literature survey in the areas of concern in each particular stage, supporting the decisions taken during this research and helping to gain confidence in the topics researched. In addition, principles of good research design explained in previous sections have been used as an input to develop the final research programme. This section provides an overview of the research programme and its formation; it also describes each stage of the research programme, including the research method selected and the reasons for each stage.

3.5.1 Overview of research programme

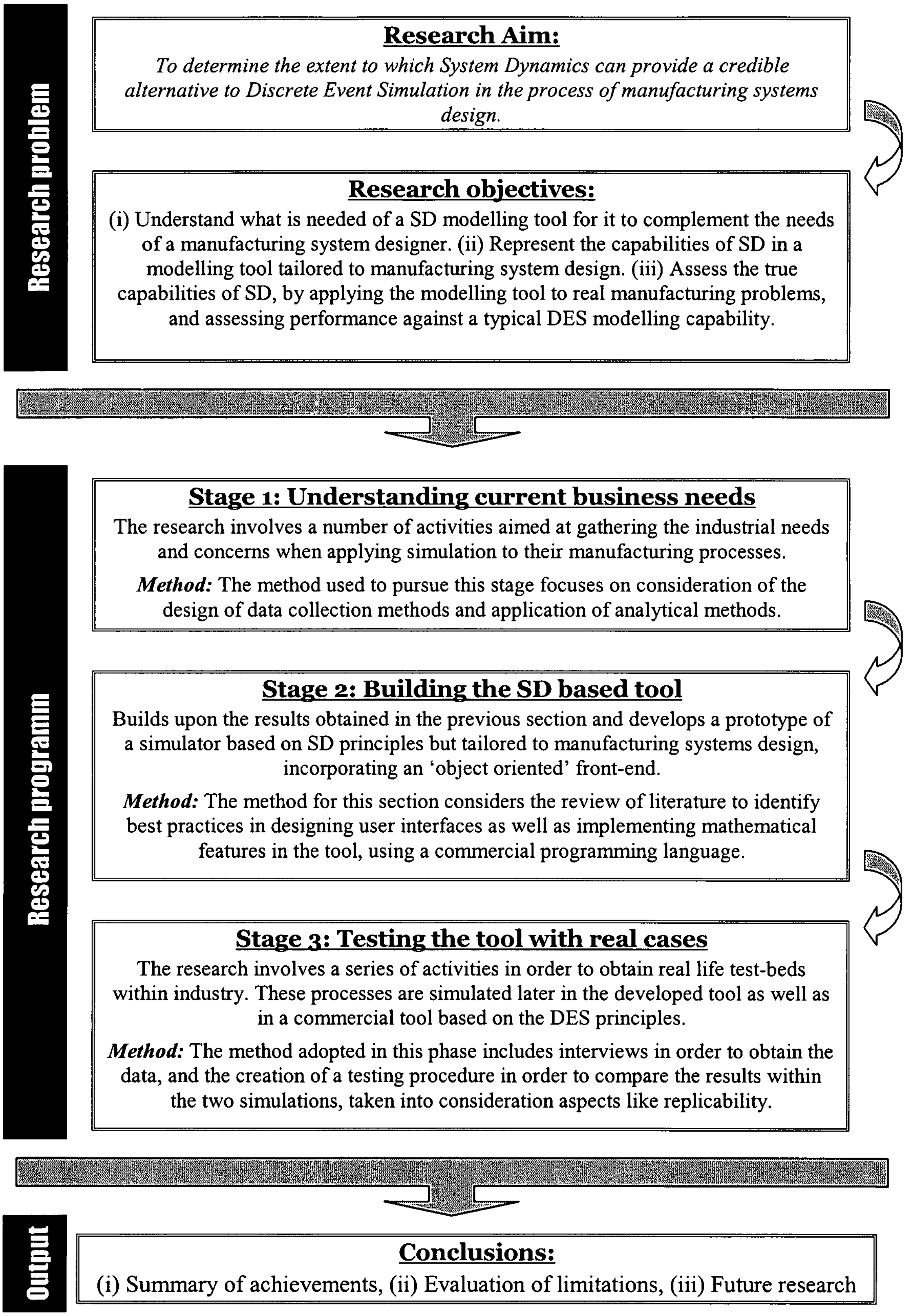
The programme designed for this research can be considered as sequential, being necessary to fulfil a stage before attempting the following one (Figure 22). However, some particular tasks within a stage can be done in parallel, and therefore, reduce the length of the research.

Stage 1 is related to the first objective of the research. It involves understanding the business needs in order to create a prototype of a tool based on the SD principles that can be applied within the manufacturing system design task (tailored for this purpose).

Stage 2 consists of developing a tool based on the principles of SD and the set of specifications gathered in Stage 1. Some assumptions are also made (see Chapter 5) in order to make the tool more user friendly.

Stage 3 is concerned with the acquisition of real test-beds and testing. The results of these tests, in addition to the literature, will lead to the discussion about the 'true' capabilities of SD when used to model manufacturing systems.

Figure 22 details the stages from the development of the research aim through to the conclusions. More detailed illustrations of each stage can be found in the following sections.



Research Aim:

To determine the extent to which System Dynamics can provide a credible alternative to Discrete Event Simulation in the process of manufacturing systems design.

Research objectives:

(i) Understand what is needed of a SD modelling tool for it to complement the needs of a manufacturing system designer. (ii) Represent the capabilities of SD in a modelling tool tailored to manufacturing system design. (iii) Assess the true capabilities of SD, by applying the modelling tool to real manufacturing problems, and assessing performance against a typical DES modelling capability.

Stage 1: Understanding current business needs

The research involves a number of activities aimed at gathering the industrial needs and concerns when applying simulation to their manufacturing processes.

Method: The method used to pursue this stage focuses on consideration of the design of data collection methods and application of analytical methods.

Stage 2: Building the SD based tool

Builds upon the results obtained in the previous section and develops a prototype of a simulator based on SD principles but tailored to manufacturing systems design, incorporating an 'object oriented' front-end.

Method: The method for this section considers the review of literature to identify best practices in designing user interfaces as well as implementing mathematical features in the tool, using a commercial programming language.

Stage 3: Testing the tool with real cases

The research involves a series of activities in order to obtain real life test-beds within industry. These processes are simulated later in the developed tool as well as in a commercial tool based on the DES principles.

Method: The method adopted in this phase includes interviews in order to obtain the data, and the creation of a testing procedure in order to compare the results within the two simulations, taken into consideration aspects like replicability.

Conclusions:

(i) Summary of achievements, (ii) Evaluation of limitations, (iii) Future research

Figure 22: Research framework

3.5.2 Stage 1: Understanding current business needs

The first stage of the research set out to identify the key features of simulation exploited in industry and the way they are modelled. The key features will be used to develop the tool in the next stage, and to define the testing methodology for the final stage of the research. It also aims to gain an understanding about the 'feelings' of simulation practitioners, software vendors and academia regarding the use of DES in manufacturing systems modelling and the reasons of the misuse of SD in the same area.

In order to obtain information about manufacturing simulation usage, suitable sources were considered. These sources are basically divided in two types, namely, (i) Primary and (ii) Secondary. Primary sources of information refer to the generation of data to address the current research problem (Tull and Hawkins, 1990), whereas secondary sources consist of information that has already been collected and has been published for reasons other than the current research problem.

In spite of the benefits of using secondary information (historical argument, collection of data, etc.), this approach is not applicable to this study, partly because of the sensitivity of the data, or the difficulty of summarising the individual preferences and behaviours of practitioners when using simulation software. Shannon (1998) reinforces the importance of individual preferences, stating that *"simulation modelling is an art that requires specialised training and therefore the skill levels of practitioners vary widely"*. Consequentially, the use of primary sources was considered as the most suitable method for information generation and collection.

Section 3.4.3 has provided a selection of data collection methods according to Robson (1993). Kinnear and Taylor (1996) simplify the categorisation of primary data collection methods into three broad areas, namely (i) Respondents, (ii) Analogous and (iii) Experimentation. Analogous situations and experimental design options have a number of inherent methodological limitations and a perceived lack of effectiveness in the context of this study. The remaining option, respondent based information generation, has the advantage of personalisation of the study and the use of the respondents' knowledge to support this research. Therefore, analogous and experimentation methods were considered unsuitable for use in this project, whereas respondent based information generation was found to be appropriate. This reasoning is supported by Drever (1995) who states that *"interviewing is one of the commonest methods used in small-scale educational research"* and Kinnear and Taylor (1996), who provide the following argument for a respondent based assessment research:

> *"When the information needs of a study require data about respondents attitudes, and perceptions, motivations, knowledge, and intended behaviour, asking people questions is essential"*

Literature provides four common methods for information collection in surveys (Ellson, 2002), namely (i) Observational techniques, (ii) In-depth interviews, (iii) Telephone interviews and (iv) Postal questionnaires. While the first technique involves the recording of the object, event or behaviour as it occurs, the other three techniques allow respondents to participate in the information generation process through questioning (Parasuraman, 1986; cited in Ellson, 2002). Although observational techniques allow a direct method of gathering data, the method was discarded for two primary reasons. First, the researcher was required to gain information from the users through the process of explanation rather than observation. Second, the magnitude of simulation projects in industry and the different steps involved with it mean that observational techniques require a great deal of time to collect data. In addition to this, observation is only possible with simulation practitioners, and so it would be difficult to include software vendors and academics.

Non-observational techniques (in-depth interviews, telephone interviews and postal questionnaires) have various strengths and weaknesses associated with them. Table 17 shows some of the most relevant characteristics of these techniques.

After considering the relative strengths and weaknesses of the three information collection methods, it was decided to select and implement the personal interview approach for this study. This opinion is supported by Voss *et al* (2002), who state:

> *"When there are questions for which no one person has all the required knowledge, or the events being studied have different interpretations or viewpoints, how and why questions may be subject to different interpretations. In such cases the researcher may consider interviewing multiple respondents."*

	Strengths	Weaknesses
In-depth interview	High levels of flexibility and control. Greater complexity and range of possible questions. Spontaneous valuable information.	Expense of the interviews Large amount of administration. Respondent anonymity problems. Inconvenience to respondents. Effects of interviewer bias on responses.
	Kidder, 1981; Dijkstra and Van der Zouwen, 1982; Kover, 1983; Brenner et al., 1985; Sokolow, 1985.	
Telephone interview	Moderately inexpensive. Reasonably reliable in maintaining respondent anonymity. Rapid collection of data. Not need too much administrative support Not as geographically sensitive as personal interviews.	Communicating complex detailed information. Interpreting answers. Required short length of interview. Constrain upon sample size and type of telephone users only. Lower threshold of control exercised by the interviews.
	Groves and Khan, 1979; Kidder, 1981; Kerlinger, 1986; Tull and Hakins, 1990.	
Postal questionnaire	Reliability in assuring respondent anonymity. Demanding a reasonably low level of administration. Possessing a high degree of standardisation. Reducing the effects of bias introduced by interviewers. Enabling the completion of questionnaires in the respondents own time. Requiring reduced resources.	Lack of control over the questionnaire completion process. Problems associated with respondents' not completing and returning questionnaires. Limited volume of data capable of being collected. Possibility of biases being present in the sample and questions asked.
	Green and Tull, 1978; Chawla et al., 1992; Paxson, 1992; Chawla and Nataraajan, 1994.	

Table 17: Strengths and weaknesses of non-observational techniques (after Ellson, 2002)

Additional rationale for justifying the adoption of the personal interview method, whilst rejecting postal questionnaires and telephone interviews, is as follows:

Postal questionnaires:

- Type of survey required to suit this method and the impossibility to know beforehand the relevancy of the respondents, added to the time required to collect the responses.
- Different interpretations of semi-structured questionnaires that might lead into misunderstandings or false conclusions.

Telephone interviews:

- Lack of communicability (both in technical aspects and language ones) between respondent and researcher.

In-depth interviews:

- Possibility of balancing the number of respondents for each segment of population (simulation practitioners, software vendors, academia).
- Control the data generation process to ensure consistency.
- Possibility to gather spontaneous information than can be useful to develop a more intuitive tool.
- Explaining the aim of the interview enables all respondents to have a similar awareness of the subject matter.

A summary of the steps used to complete Stage 1 is illustrated in Figure 23.

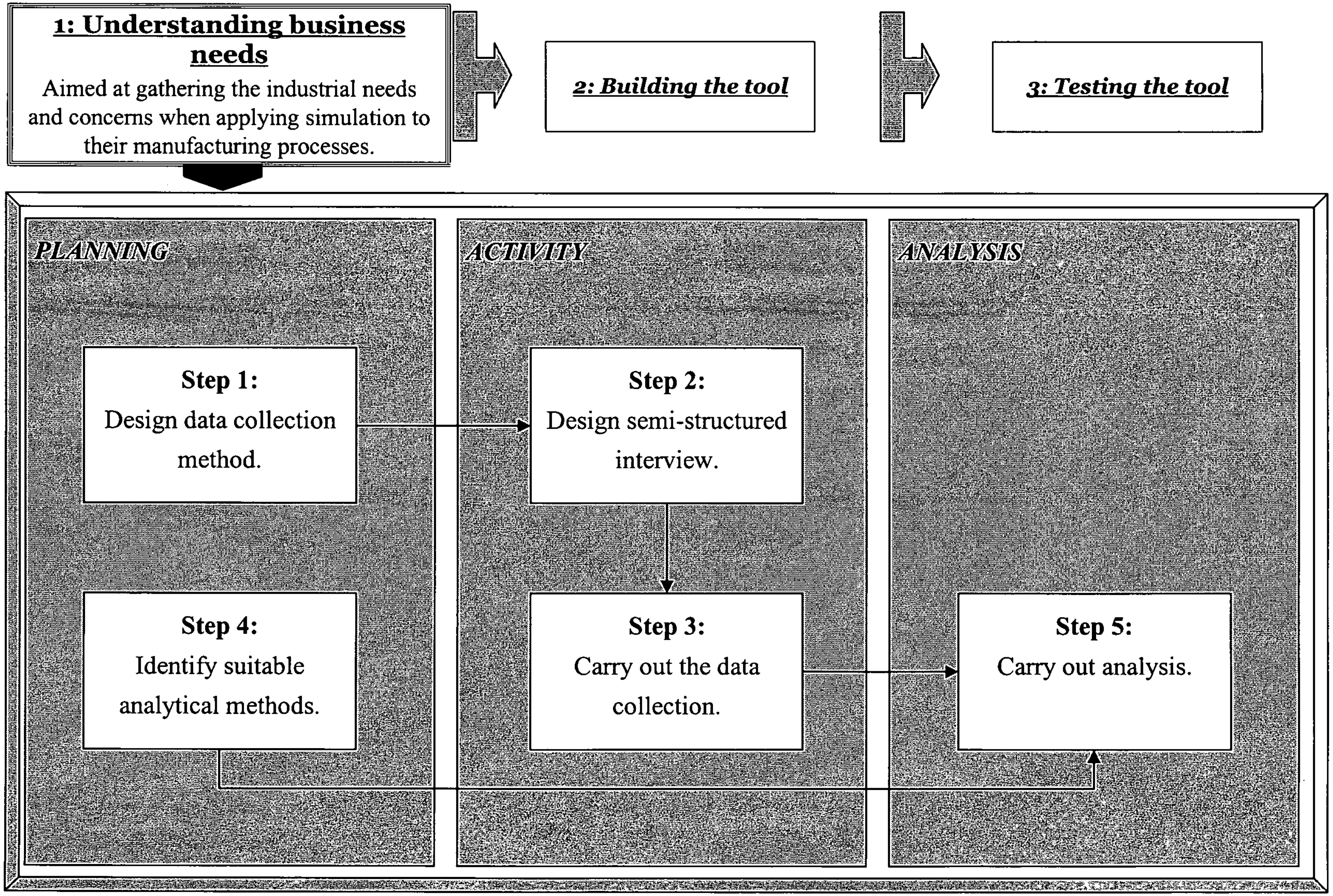


Figure 23: Summary of steps undertaken in Stage 1 of the research programme

3.5.3 Stage 2: Building the tool based on SD principles and business needs

Stage 2 of the research is developed from the findings obtained after the completion of Stage 1 (see Section 3.5.1). This part of the research answers the second of the key objectives stated in Section 3.2: *"Represent the capabilities of SD in a modelling tool tailored to manufacturing system design"*. Stage 2 can be completed by dividing it into two main tasks, the first being an investigation and definition of the specifications that the tool will include (based on interviews carried out in Stage 1, and analysis of existent software), and second, development of the tool and verification of its features.

3.5.3.1 Building the specifications of the tool

Upon completion of the first stage of the research, a picture will emerge of the features that practitioners, academia and software vendors consider essential in any commercial simulation software. Other features will be considered more or less dispensable, although their inclusion might enhance the overall functionality of the tool. Furthermore, there should be information about the kind of features that an 'intuitive' tool should avoid (e.g. 'real' numbers instead of 'integers' when referring to 'parts' produced). However, there is unlikely to be detailed information about the requirements of the tool in terms of usability and user interface design.

This concern leads into the analysis of existing simulation tools based on the DES technique because, as stated in Section 2.4, *"A strong reason for using Discrete Event Simulation within the manufacturing industry is the personalisation of those tools to deal with manufacturing processes."* A tool based on the SD technique also needs to be analysed in order to help identify the main differences between these simulation tools. The analysis will also help to understand the importance of the existing tools in the use of one or the other technique when simulating manufacturing processes.

These two inputs (interviews results and analysis of existent software) can be used to build the final requirements of the software. This is a critical phase due to the importance of the functionality of the tool when testing real-life cases in Stage 3. A lack of functionality in basic features can make the tool impossible to use in industry, whereas an excessive inclusion of features can result in time delays and therefore be inappropriate for the aim of this research.

3.5.3.2 Building and validating the tool

Guidelines or appropriate methods applicable in programming software can be divided into two areas (Knuth, 1998), namely: (i) Generic concepts (or non-dependent of the programming language adopted) and (ii) Tool specific (those that depend on the chosen tool).

A number of authors (Gamma, 1995; McConnell, 1996; McConnell, 1997; Fowler and Scott, 1999) suggest a rigorous working method when developing 'computer code' and describe procedures to accomplish this task. Royce (1998) and Wideman (2003) suggest 'the waterfall life span', which is basically a linear process, as a valid method for building software. Due to its simplicity and adequacy, the waterfall approach has been used during this research. Duncan (2000) takes this concept further by developing a 'spiral' that detail all the main events, which are categorised into four main groups of activities, namely: (i) Identify, (ii) Design, (iii) Construct and (iv) Evaluate. These two approaches can be seen in Figure 24 and Figure 25, respectively.

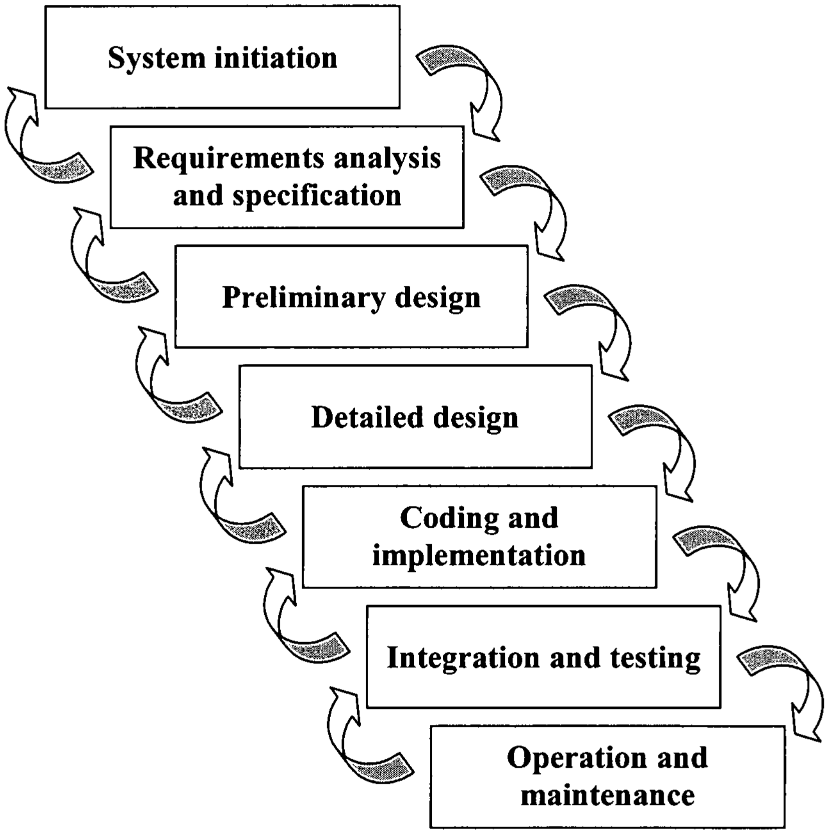


Figure 24: The waterfall life span (Royce, 1998)

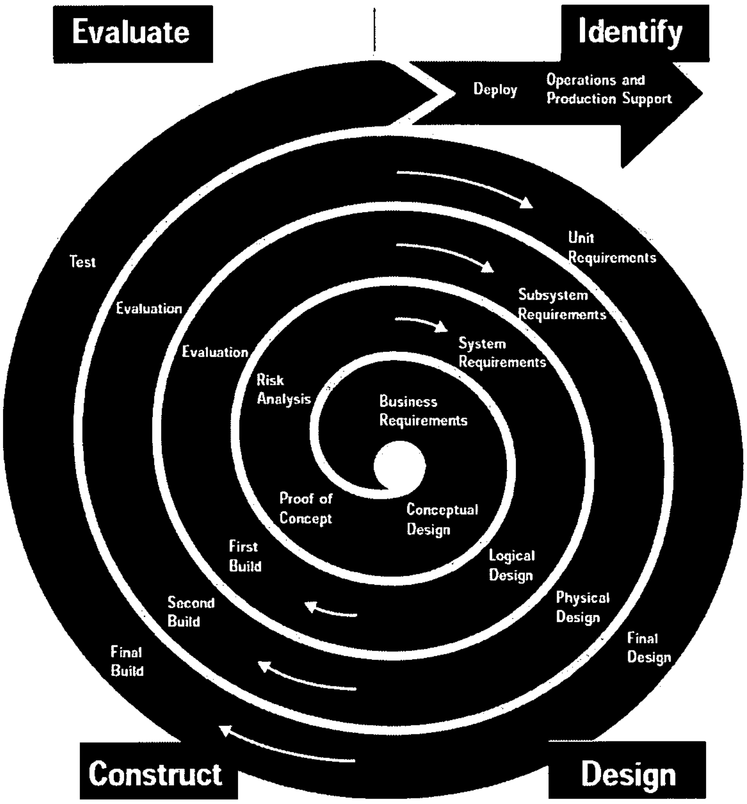


Figure 25: The software development life (Duncan, 2000)

The final usefulness of the tool depends not only on the technique selected, but also on the programming language selected. Chapter 5 provides a detailed discussion of the procedure adopted for tool selection.

In order to obtain reliable results, the tool needs to be validated by testing it against known conditions so that its behaviour can be analysed. This phase tests the accuracy of the results for the developed tool when given a specific set of input data. Validity refers to the degree to which a measure assesses what it purports to measure (Fink, 1998). In this context, validation of the software comprises both the verification and testing of its features (Ellman, 2000), which is done in two stages, as follows:

1. **Interface:** Where all links and functions of the interface are executed to check its suitability with the predefined functionality.
2. **Implemented functionality:** Where all functions implemented in the code are verified in the first instance, and tested with known input values to check the correctness of the output values.

Figure 26 summarises the steps followed to complete Stage 2.

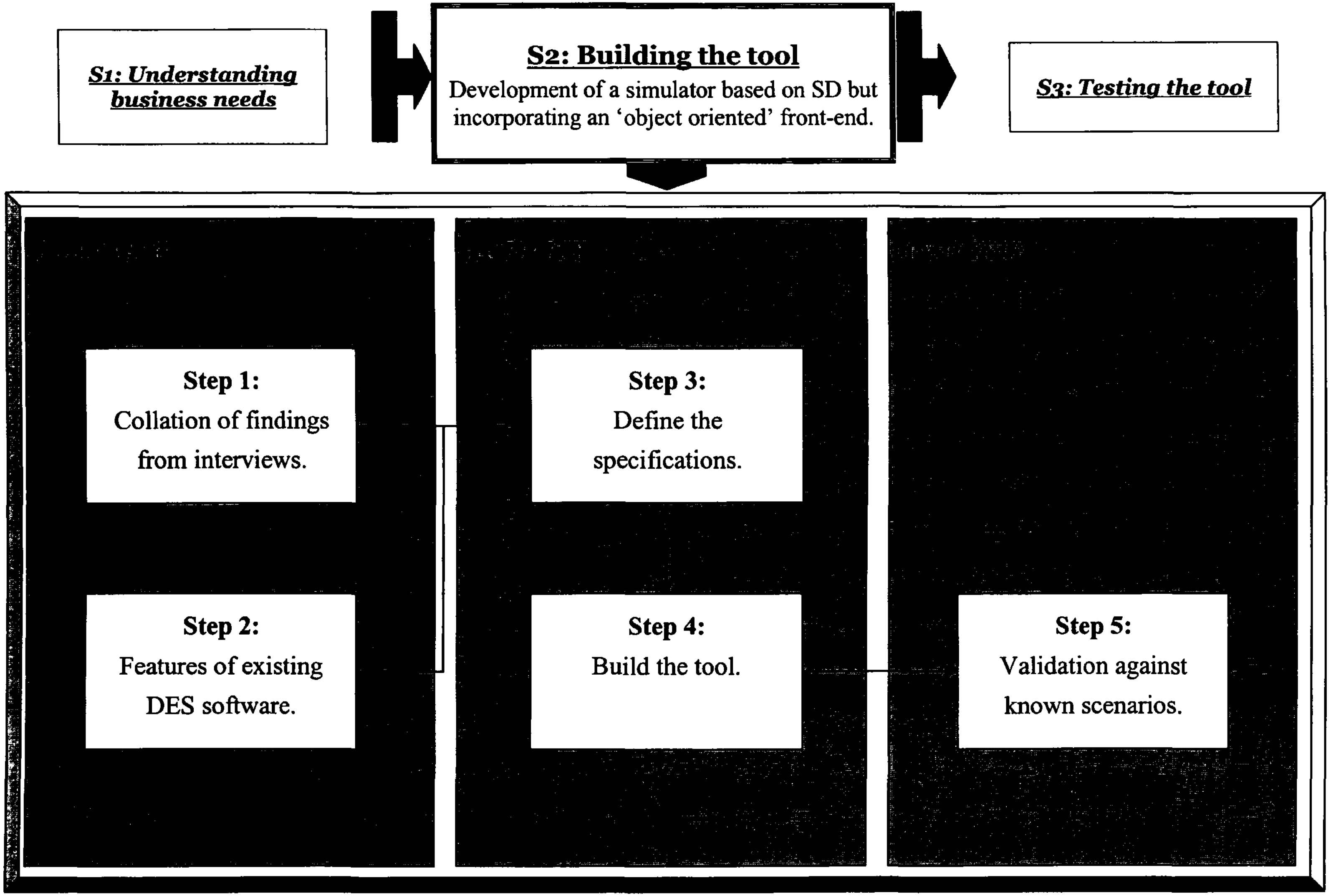


Figure 26: Summary of steps undertaken in Stage 2 of the research programme

3.5.4 Stage 3: Testing the tool with real cases

The final stage of this research is carried out following the completion of the previous stages. It attempts to clarify the usefulness of both SD and DES when modelling different manufacturing scenarios, by accomplishing the third objective of this research, which is to:

> *"Assess the true capabilities of SD, by applying the modelling tool to real manufacturing problems, and assessing performance against a typical DES modelling capability."*

3.5.4.1 Obtaining industrial 'Test Beds'

This task involves gathering effective cases from companies, analysing their processes, and verifying that they meet the requirements of the research. Case studies and their number can vary in size depending on factors that include finance availability, time, number of researchers, geography and so on. Single case studies are often in-depth and longitudinal; however no clear definition exists of what a single case study actually is (Voss *et al*, 2002). Weaknesses of single case studies include the limits to generalisability of conclusions and avoiding bias. The alternative is the use of multiple cases, where depth of study may be reduced but it can help to guard against observer bias and add to external validity (Voss *et al*, 2002). Multiple cases would appear to better suit the requirements of the research study. With multiple case research, the sample of cases is selected using some set criteria and not sampling (Eisenhardt, 1989; Yin, 1994). Eisenhardt (1989) appears to be the only researcher to suggest a guide as to the number of cases that a typical study should conduct:

> *"A number between four and ten usually works well".*

For the reasons highlighted above, in conjunction with the aim of the research, the use of the multiple cases strategy appears to be the most appropriate data collection method for the final stage. These test beds are obtained by contacting relevant companies and analysing their processes in order to establish whether they fulfil the features that are to be analysed. After analysing a number of companies, the most appropriate are selected.

3.5.4.2 Performance analysis of modelling techniques

In this phase, the suitability of each technique in different situations is assessed. This suitability will be defined by considering practitioners' expectations as well as the effort required. When both methodologies deal with a particular set of requirements, different features are balanced in order to identify the strengths and weaknesses for each technique.

This stage requires an experimental design that allows all features to be tested correctly. When developing this experimental design, special effort is needed to isolate those features that need to be tested separately. Then, the experiments will be made both in the developed tool as well as SD and DES based commercial software in order to analyse the differences related to both the technique and the tool. The order in which the tests are made is also critical, because there is a learning curve (in some cases, replicability of test is only possible for results, and not procedures). To avoid this problem, tests will be developed by different people in a different order, and differences will be analysed.

Figure 27 summarises the steps followed to complete Stage 3.

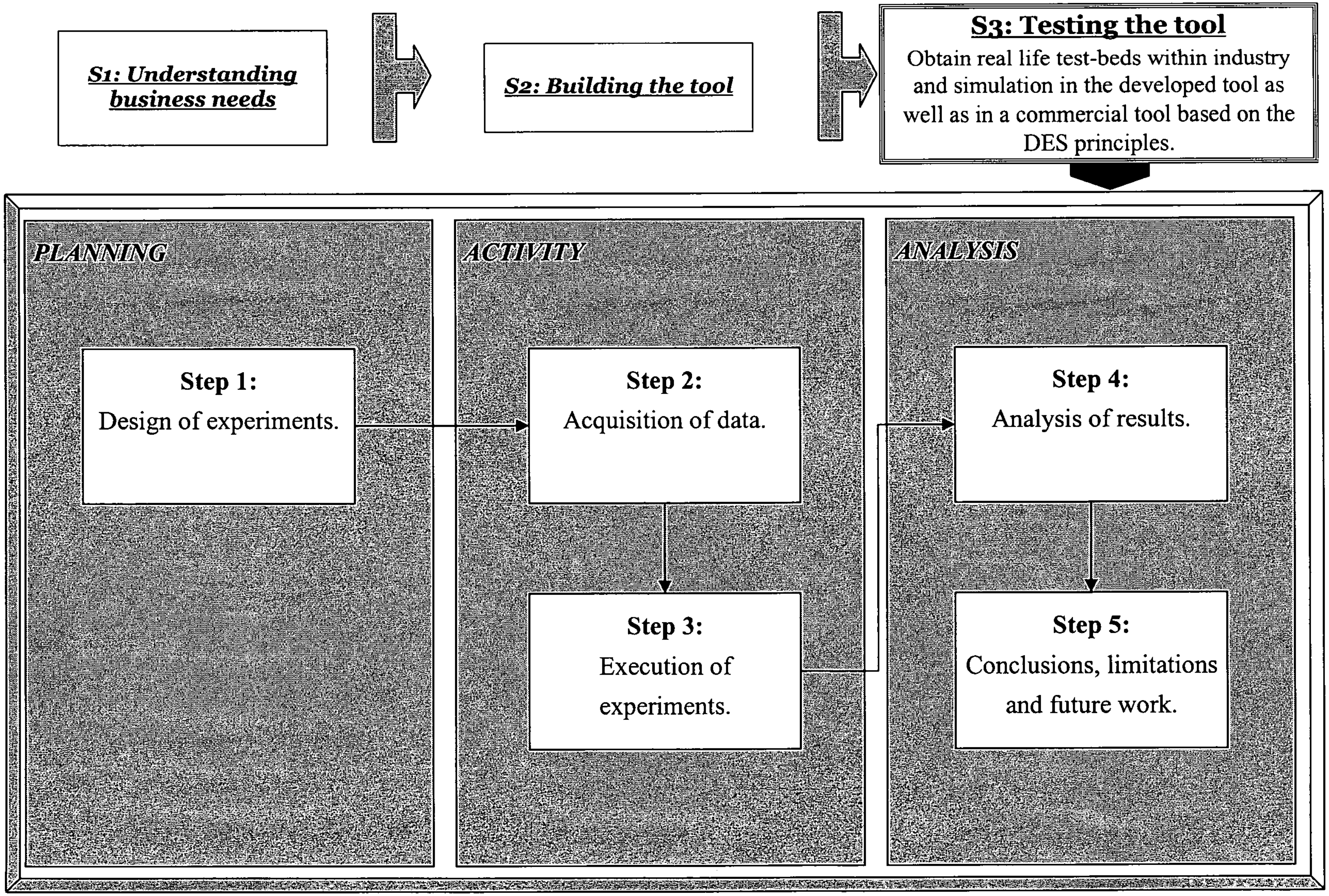


Figure 27: Summary of steps undertaken in Stage 3 of the research programme

3.6 Summary

This chapter has set out the intended plan of how the research aim and objectives are to be attained, through addressing the research objectives. The literature study highlighted apparent gaps in the research area, specifically the lack of usability of SD when applied to manufacturing systems modelling. A possible explanation for this misuse is that existent SD tools are not tailored for this task. The most appropriate research strategies that address this requirement are developed. Three separate but linked stages have been developed. The first stage uses an interview research strategy to address the first research objective. The second stage addresses the second objective by developing a computer based tool programmed using Visual Basic and adopting some functionality of the SD technique. Finally, the last stage addresses the third objectives of this research by testing and analysing the effectiveness and usability of the tool for a given number of scenarios. These results lead into the conclusions and limitations of this research. These stages collectively form the overall research programme that is illustrated in Figure 22.

Chapter 4

REQUIREMENTS OF A SYSTEM DYNAMICS MODELLING TOOL FOR MANUFACTURING SYSTEM DESIGN

Chapter 2 analysed the state of simulation within industry, trends, and a gap between the exploitation of DES and SD. From this, the research aim and programme were defined. This chapter describes the first stage of the research, which addresses the first research objective by surveying a selection of companies and academia. It also describes the design and content of the survey used, and identifies the necessary features for a SD based simulator tailored to manufacturing system design. The structure of this chapter is as follows:

1. Methodology followed to design the survey
2. Questionnaire design.
3. Survey execution and results.
4. Analysis of survey results and key findings.

4.1 Methodology for survey design

In order to truly compare the characteristics of SD and DES within the process of manufacturing system design, it is first necessary to gain a clear understanding of the needs of practitioners who use simulation techniques. The first stage of the research, described in this section, presents the design of this survey. Section 3.5.2 revealed that the most appropriate method for achieving this is a small, but in depth, survey of key and experienced simulation users from industry and academia. However, the format of the personal interview has to be considered. The personal interviews need to be structured to ensure a common respondent knowledge level, consistency, and an avoidance of bias (Section 3.5.2). This section presents the design of this survey.

Section 3.4.3 and Section 3.5.2 suggest that a semi-structured interview is the most appropriate method to follow for this particular research. Thus, the questionnaire design for this survey contains both 'closed' and 'open' questions, which in the view of Eisenhardt (1989) can be highly synergetic. This questionnaire is the same for each respondent and utilises the same definitions exposed in the presentation given before the interview. Thus, the respondents answer a set of 'closed' questions that are regulated and consistent. In the second part of each section of the questionnaire the respondents are asked for their views in a semi structured style, thus enabling data collection on an individual and more exploratory basis.

The personal interviews, therefore, consist of a brief presentation, followed by a questionnaire consisting of five sections (explained in Section 4.2 and presented in Appendix D), each of them containing both 'closed' questions and a debate. These questions were designed for use in a supervised environment (the researcher was always present during the interview), in which the respondents had sufficient knowledge of the subject area.

4.1.1 Survey sample

As stated in Section 3.5.2, three different industrial sectors were targeted: (i) Simulator practitioners from various industry sectors, (ii) Simulation software developers and (iii) Members of academia with experience in simulation. Section 3.5.2 also recommended a small quantity of in-depth interviews as an appropriate survey approach. In total, fourteen people were interviewed from the previously described areas. In order to ensure a minimum level of expertise in this area, special care was taken when selecting simulation practitioners. In addition, the number of respondents for each sector were balanced to specially weight practitioners in respect to academia and software developers because they are the targeted users of the simulation tool. Due to the quick response that this technology provides, electronic mail was considered as a suitable method for approaching interviewers. This email letter included a brief presentation of the research, its objectives and aspects related to the interview.

4.2 Questionnaire design

The design of the interview questionnaire was closely related to addressing the needs of practitioners when using a simulator based on continuous methods, by analysing their actual needs and thoughts. The presentation letter can be found in Appendix C and the interview questionnaire itself is presented in Appendix D. Care was taken to set and ask relevant questions in such a way as not to bias respondents' answers or allude to an answer. Recommendations from Robson (1993), Oppenheim (1992) and Foddy (1993) were followed when conducting the interviews, which included interview style and procedures to avoid. The main part of the questionnaire is designed in 'semi-open' questions; that means, although a range of options do exist for answering the questions, comments or beliefs can be made to justify the answer of the respondent (e.g. two respondents might consider that 'step by step' analysis is essential, but their reasons for using it might be different).

The objective of each section of the questionnaire was as follows:

- **Section 1**: This section evaluates which 'elements' and 'scenarios' the interviewees consider more relevant. The proposed elements are taken after the analysis of several DES simulation packages (Section 2.4.3). In addition, possible alternatives for substituting specific elements (for example, conveyors) for simpler ones (for example, buffers) were explored.
- **Section 2**: This section identifies and evaluates where interviewees spend their time when simulating manufacturing processes, as explained in Section 2.2. Its objective is to gather information on the relative importance of each task, in order to invest more effort (when developing the tool) in those areas where they provide more benefit to industrialists.
- **Section 3**: This section identifies and evaluates the main results that practitioners obtain from the models they simulate and the importance of them. It also aims to gather information on the 'quality' of the results they expect to obtain. The performance measures included in the questionnaire have been chosen because of their relevance (Oyarbide, 1999; Okudan and Kabadayi, 2001) and the possibility of being implemented during this research.
- **Section 4**: This section includes questions for gaining knowledge about the use of simulation packages. Thus, it evaluates how interviewees analyse their models while they are developing them and the way they present their results.
- **Section 5**: This section evaluates the importance of the human and technical resources used by the practitioners when simulating processes.

The questionnaire is also set to fit the time frame available in the industrial environment. Although the researcher considered taping the interviews, a decision was taken not to do so. The decision was based primarily on the fact that the interviewees might have been less responsive if a tape recording had been made. The questionnaire development process, following the recommendations of Churchill (1998), can be observed in Figure 28.

Prior to the visits, information regarding the research was forwarded to the main contact person in the company. The interviews were also preceded by a short presentation, which introduced the research and explained the purpose of the visit. This presentation aimed to assist respondents in their understanding of terminology and concepts applied within the questionnaire. The objectives of the questionnaire and each of its five modules were also provided in the questionnaire and are summarised in Section 4.2.1. After guaranteeing their confidentiality, the interviewees were individually interviewed by the researcher. The researcher recorded the responses given during the interview.

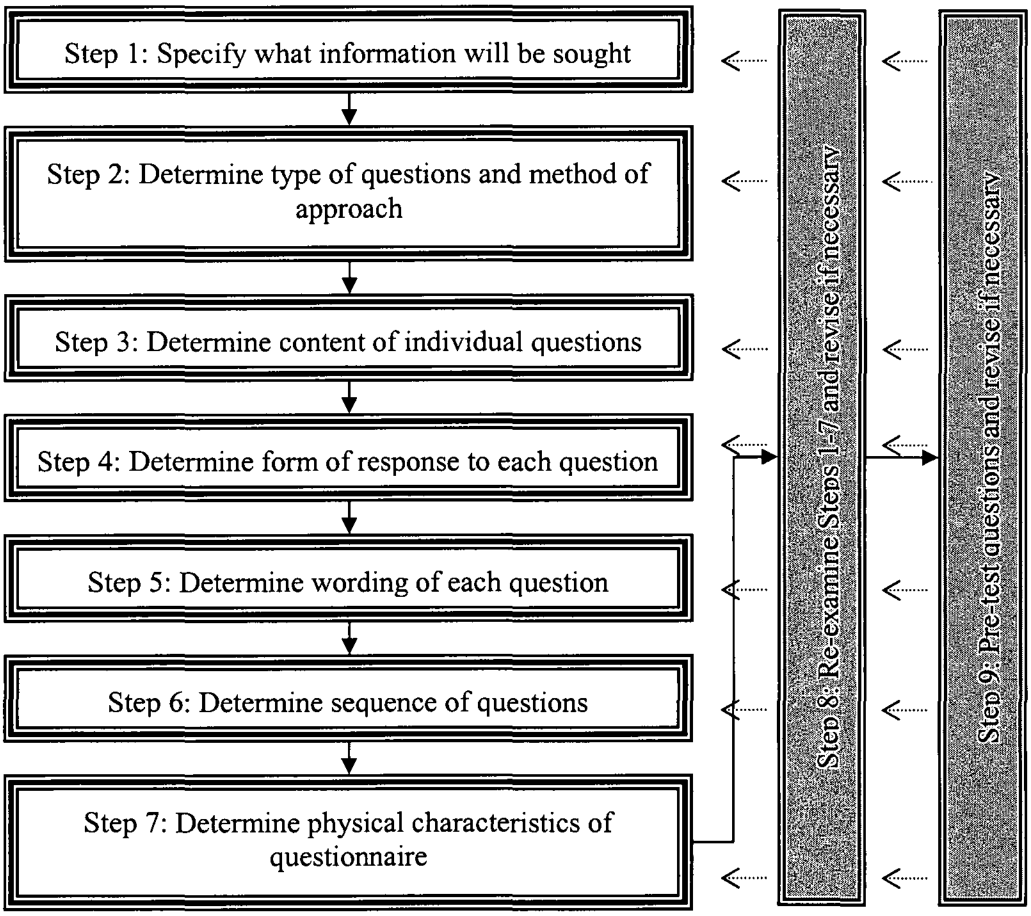


Figure 28: Questionnaire development process (after Churchill, 1998)

As illustrated in Figure 28, the questionnaire development process involved the following:

Step1 – Information sought: The first stage of developing a questionnaire requires the determination of boundaries within the study and a clear understanding of the information sought. This survey is conducted to explore how industry applies simulation tools (which are the inputs, elements chosen to simulate them, outputs provided, conducted process and invested and resources)

Step 2 – Type of questionnaire and method of administration: This stage describes how the information sought should be collected. A structured questionnaire provides some of the required information and provides a framework for a detailed and quantifiable basis for analysis.

Step 3 – Individual question content: Questions relating to current practices and context were specifically targeted within the questionnaire.

Step 4 – Forms of response: Once the questions were formulated, a suitable method for recording the answers was investigated. In the case of 'closed' questions, a formal criterion of measurement and scale was constructed in order to assign numbers to objects (Nunnally, 1967). Similar to the Quality Function Deployment (QFD) technique, a numeric value was given to 'essential', 'important' and 'avoidable' responses when constructing the graphs. These values are, respectively, '3', '1' and '0'. Due to the relatively small number of respondents and the scale chosen for measuring the answers, statistical analysis was not considered to be necessary.

Step 5 – Question wording: The questions were designed so that they were simple, unambiguous in their wording and the language used, neutral.

Step 6 – Question sequence: The questions were arranged so that the broader and less difficult to answer questions precede those requiring more thought (Churchill, 1998) (see Section 4.2.1 for a more detailed explanation).

Step 7 – Physical questionnaire: In this research, the respondents had already expressed an interest in the study and they had taken time to arrange and participate in the interviews. Therefore, the questionnaire concentrated on functionality and ease of use, rather than visual appeal.

4.2.1 Determination of question sequence

The questionnaire was designed for use in a supervised environment in which the respondents had a previous knowledge about the area in question. It is also structured in five sections in order to organise the interviews, and to gradually increase the complexity of the topic. It covers: (i) The simulation standard elements used when creating a model, (ii) The steps of a simulation project, (iii) Results obtained from the models, (iv) Type analysis carried out and (v) Resources involved. The first section of the questionnaire is aimed at evaluating of the 'elements' and 'scenarios' the interviewees consider more relevant. The second section targets the understanding of the way practitioners simulate by gathering the time they spend in each stage of the simulation process. The third section deals with the main results obtained from the simulation models and the importance of them. The fourth section evaluates how interviewees analyse their models while they are developing them and the way they present their results. Finally, the fifth section gathers information on the human and technical resources used by the practitioners to successfully complete the project in hand. Each section starts with a brief explanation followed by the questionnaire. A special effort was made to obtain comments from the interviewees without interfering with their responses.

4.2.2 Determination of questions type

The questionnaire uses both 'closed' or pre-coded answers aided by 'open' or free-response types of question. A 'closed' question is one in which the respondents are offered a choice of alternative replies. This type of question allows less freedom of expression, but they are easy to answer and analyse. Conversely, 'open' or free-response type questions are not followed by any kind of choice, and the answers have to be written in full. These questions allow freedom of expression and are easy to ask, but might be difficult to answer and analyse (Gillham, 2000a). This stage of the research is more exploratory in nature and therefore more difficult to qualify with a quantitative approach to analysis (Eisenhardt, 1989; Hamel *et al.*, 1993; Denzin and Lincoln, 2000 and Sanjek, 2000). Open questionnaires often involve personal emotions and biases (Roy, 1997).

4.2.3 Analysis of responses

For each question in the questionnaire, the responses given by the respondents are compiled into two categories. The first category identifies the 'close' questions answer provided by the respondents. Thus, the respondents answer a set of questions that are regulated and consistent. Conversely, the second category includes the 'open' questions, such as comments or thoughts. This uses a semi structured style to enable data collection on an individual and more exploratory basis. Unless otherwise stated, all the responses listed in the subsequent sections of this chapter are derived from the majority of respondents. Specific reference will be made to any other responses made by such people as practitioners, academics or software developers.

4.3 Survey results

The results obtained from the survey are presented in this section. The presentation of the results mirrors the layout of the questionnaire itself. Following the presentation of contextual data, each question from the questionnaire is presented along with associated results. Table 18 lists the fourteen organisations involved in this survey, together with their industry sector and location. The following diagrams summarise contextual information about the organisations that participated in this study, the survey sample.

No	Company	Industry sector	Location
1	**A**	Automotive	West Midlands, UK
2	**B**	Automotive	West Midlands, UK
3	**C**	Automotive	Essex, UK
4	**D**	Automotive	Sunderland, UK
5	**E**	Aeronautics	Durham, UK
6	**F**	Academia	Warwickshire, UK
7	**G**	Academia	West Midlands, UK
8	**H**	Software developers	Worcestershire, UK
9	**I**	Consumer goods	Aretxabaleta, Spain
10	**J**	Consumer goods	Oñate, Spain
11	**K**	Consumer goods	Arrasate, Spain
12	**L**	Academia	San Sebastian, Spain
13	**M**	Academia	Arrasate, Spain
14	**N**	Software developers	Arrasate, Spain

Table 18: List of survey sample organisations

Figure 29 highlights the origin of the companies visited. This research is funded by the Basque Government, who considered it relevant to extend the research to that geographic area. It was also considered important to include Basque companies because it could help the researcher to make contact with companies that could participate in the next stage of the research (obtaining the test-beds). It was also considered beneficial because the type of companies within Basque Country are mostly SME's; a study done by EUSTAT (2002) revealed the existence of about 800,000 working people distributed in more than 150,000 companies. This peculiarity, added to the fact that there has been little implementation of simulation in the Basque Country, might offer different views concerning simulation than the companies visited in the UK (mostly large companies). Thus, six of the fourteen companies visited were located in the Basque Country.

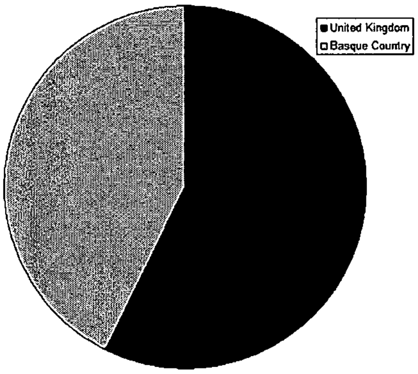


Figure 29: Number of firms vs. country

As stated in Section 4.1.1, special attention was made to balance the number of different respondents of each area. The exploratory nature of this survey justified this procedure. Figure 30 highlights this concept and shows the final balance between the three different sectors targeted in this survey. In total, eight people from industry, four members from academia and two software developers were interviewed.

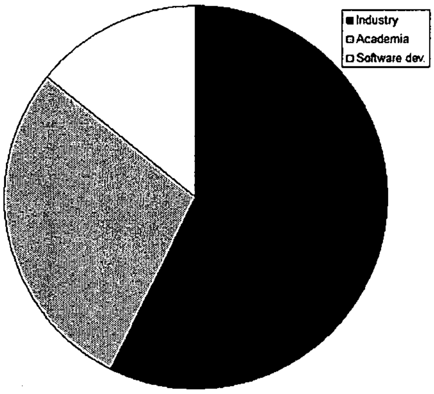


Figure 30: Areas of sampled population

Figure 31 shows in more detail the specific sectors targeted within industry and their size. The figure on the left depicts the different sectors covered by this survey. These were selected after considering the product they manufactured and their size. For example, while both the automotive and aeronautics industries tend to produce products of a relatively high value, the automotive industry produces them at a much higher rate. Conversely, companies visited from the consumer goods sector produced elaborate products of much lower value, with the cost of materials being a less critical factor than the previously mentioned sectors. This argument, amongst others considered, justifies the selection of these companies when carrying out this survey.

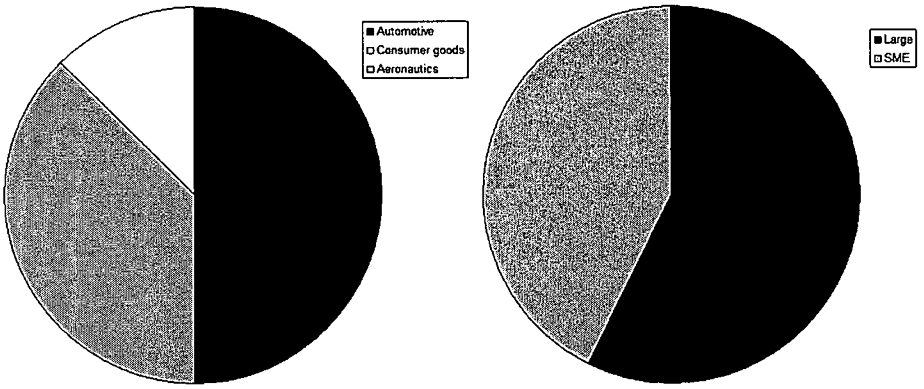


Figure 31: Industry sectors and companies size

4.3.1 Section 1: What do practitioners simulate / what a manufacturing simulator based on SD should simulate

This section of the survey asked what the main elements were that practitioners include when simulating their models and which were the minimum requirements that the tool developed in this research should include in this matter (those elements answered as 'essential'). It also inquired as to the detail of the type of flows they simulate (in terms of 'elements' connections). Finally, accessory elements were evaluated and rated by respondents in order to detail the framework or environment of the new tool. It is important to highlight that a level of abstraction was required by respondents to extrapolate their practices (they were from different sectors and used different tools) to the usability and scope of the new tool (through comments or 'open' responses). The following tables and graphs summarise the responses given by interviewees. It also quotes some of the main comments gathered during the interviews.

4.3.1.1 Model elements

A model's elements are concerned with the most common elements found in actual DES based tools. Table 19 and Figure 32 summarise the 'close' responses and accumulative weight given by the respondents in this sub-section of the questionnaire. Some quotes from respondents are given below:

DESCRIPTION	RATES													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Part	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Buffer	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Machine	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Labour	○	⦿	○	●	○	○	○	○	○	○	N	○	○	N
Conveyor	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Shift	●	⦿	⦿	○	○	⦿	○	○	○	○	○	○	⦿	N
Vehicle	⦿	○	○	○	○	○	○	○	○	○	○	○	○	N
Track	⦿	○	○	○	○	○	○	○	○	○	○	○	○	N

Note: ● = Essential; ⦿ = Important; ○ = Avoidable; N = Not answered.

Table 19: Importance of model's elements

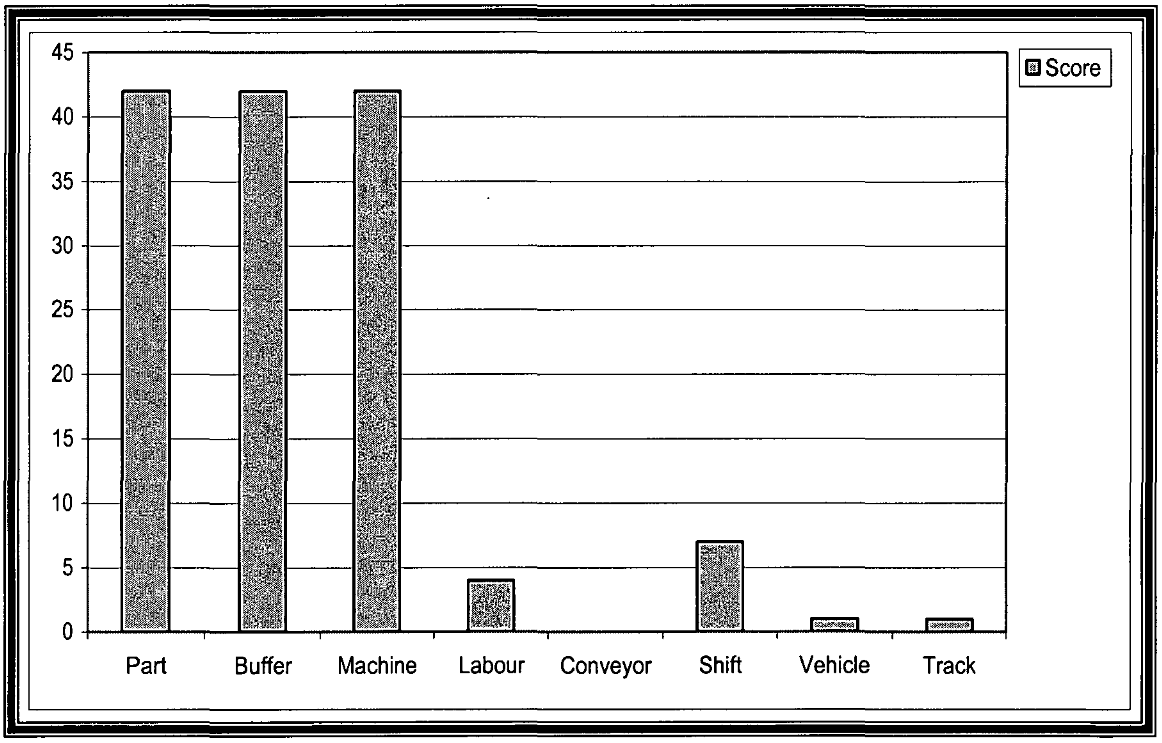


Figure 32: Accumulative importance of model's elements

- **Respondent A:** *"Labour and conveyor can be simulated using machines".*
- **Respondent C:** *"I don't like to see 'real' numbers when dealing with products. The tool should provide 'discrete' numbers instead".*
- **Respondent E:** *"Approximate stock levels to the next integer value."*
- **Respondent F:** *"Entities and attributes might be difficult to simulate in SD" "Labour can be simulated by delays".*
- **Respondent H:** *"Parts, buffers and machines are enough."*
- **Respondent I:** *"Materials flows seem confusing. I prefer the 'discrete factory'"*
- **Respondent M:** *"Try to reduce the number of elements to the essential ones"*

4.3.1.2 Model flows

In order to create a path where elements can flow through, elements contained in a model need to be linked. These can be in various forms, from the most simple 'transfer line', where all elements flow through a single path in single file, until the most complex flows where elements can follow different routes depending the conditions of the system and machines can be linked to/from N different machines (N being a positive integer number). Table 20 and Figure 33 summarise, respectively, the responses and accumulative weight given by the respondents in this sub-section of the questionnaire. Some quotes from respondents are also given below.

DESCRIPTION	RATES													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1 to 1	●	●	●	●	●	●	●	●	●	●	●	●	●	●
1 to N	●	●	●	●	●	●	●	●	●	●	●	●	●	●
N to 1	●	●	●	●	●	●	●	●	●	●	●	●	●	●
N to N	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Batch	●	●	●	N	○	●	○	⦿	○	⦿	○	○	○	N

Note: ● = Essential; ⦿ = Important; ○ = Avoidable; N = Not answered.

Table 20: Importance of model's flows (connections)

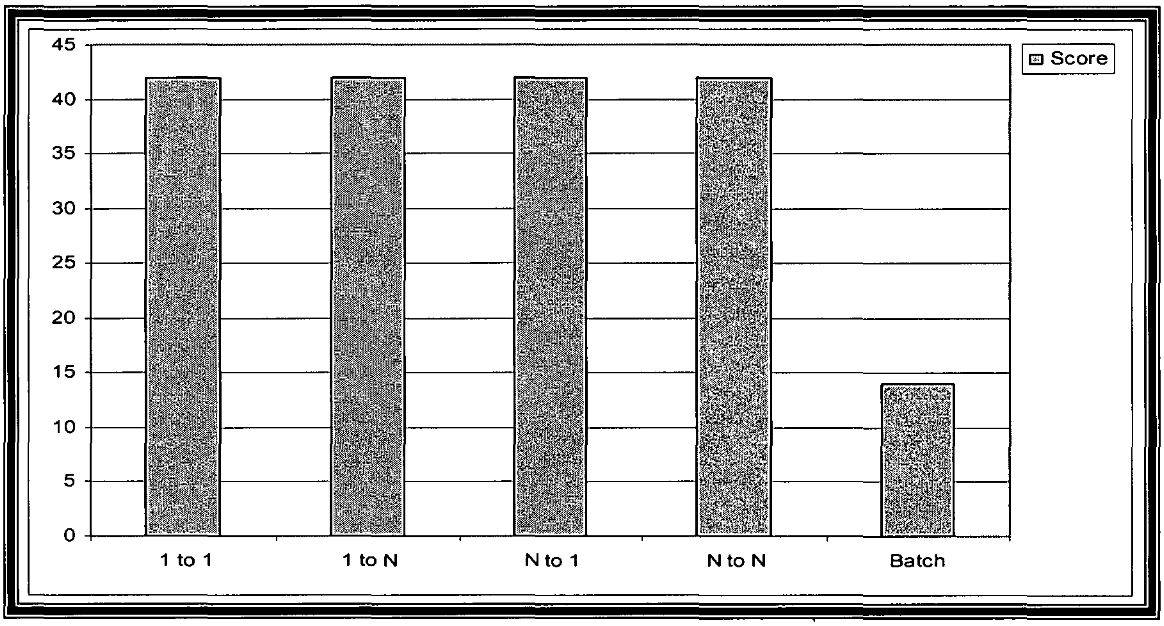


Figure 33: Accumulative importance of model's flows

- **Respondent A:** *"If the tool does not include N to N flows, it is non useful in industry".*
- **Respondent C:** *"If different products follow the same flow, sometimes it is possible to simulate one by one, execute them in different runs, evaluating and combining the solutions in a final stage".*
- **Respondent F:** *"Materials might come in batches, but a SD model will treat them as levels".*
- **Respondent G:** *"Analyse the necessity of simulating batches, they might not be required".*
- **Respondent L:** *"SD 'machines' can include prioritisation of inputs, so certain types of products can be developed earlier than others".*

4.3.1.3 Model accessory elements

Models are not built using only basic elements and flows. In many cases, they also require auxiliary elements to deal with, for example, the casual events of the system or even store the model's information. Causal events are often modelled using random numbers, breakdowns or setups, while information can be stored as constants or variables. This sub-section aims to collect the importance of these elements for the interviewees. Table 21 and Figure 34 summarise, respectively, the responses and accumulative weight given by the respondents in this sub-section of the questionnaire. Some quotes from respondents are also given below.

DESCRIPTION	RATES													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Constant	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Variable	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Random number	●	⦿	⦿	N	○	○	○	⦿	⦿	N	N	○	○	N
Breakdown	⦿	⦿	●	●	⦿	●	●	●	⦿	●	●	●	⦿	⦿
Setup	⦿	⦿	●	●	⦿	●	●	●	⦿	●	●	●	⦿	⦿

Note: ● = Essential; ⦿ = Important; ○ = Avoidable; N = Not answered.

Table 21: Importance of model's accessory elements

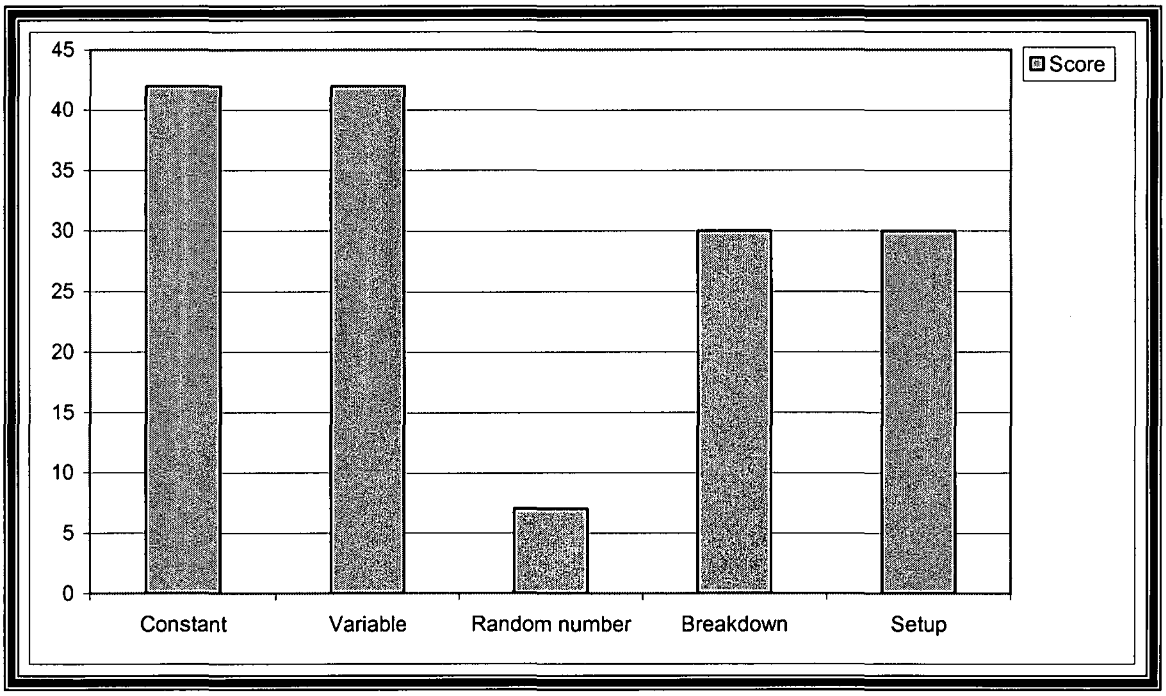


Figure 34: Accumulative importance of model's accessory elements

- **Respondent A:** *"If breakdowns are simplified, analysis like 'if this machine stops, how much time do I have?' cannot be done".*
- **Respondent C:** *"Breakdowns and changes are essential. They can be simplified to percentage values".*
- **Respondent M:** *"Random numbers and SD are incompatibles".*

4.3.2 Section 2: How practitioners simulate / how should they simulate

This survey section goes beyond the processes and elements required by industry to create a model of their systems. It aims to discover the complexity of the different tasks involved in the modelling and simulation process by measuring the time required for carrying out these tasks. Literature suggests that common tasks involved in a simulation project are: (i) Data collection, (ii) Model coding, (iii) Model verification, (iv) Model execution, (v) Model validation, (vi) Model experimentation, (vii) Model expansion and (viii) Analysis of results (Section 2.2). 'Time' is the variable chosen to measure complexity because of its simplicity and suitability with the scope of this survey, which focuses on gathering information to create a sensible set of specifications based upon industrial needs.

4.3.2.1 Time consumed when modelling and testing

Table 22 summarises the responses given by respondents regarding the required time to complete each simulation stage. Some quotes from respondents are given below:

DESCRIPTION	RATES													
	A	B	C	D	E	F^2	G	H	I	J	K	L^1	M	N
Data collection	N	N	19	N	N	40	N	10	23	N	N	20	N	N
Model coding			19					50	9			20		
Model verification			12						6					
Model execution			8			20		10	3			20		
Model validation			3						9					
Model experimentation			19			30		20	43			30		
Model expansion			8						1					
Analysis of results			12					10	6			10		
Note: Results converted into percentage values. N = Not answered. (1) = Ideally, (2) 10% left = Documentation.														

Table 22: Time consumed during the different stages of a simulation project

- **Respondent A:** *"Data collection should be reduced as much as possible".*
- **Respondent B:** *"Data collecting and validating is the most time consuming".*
- **Respondent C:** *"If the final user is not going to be a simulation expert, tools like Microsoft Excel can be useful to code a model. In any case, it must be intuitive".*
- **Respondent E:** *"Simple and efficient graphics might help to model execution and validation".*
- **Respondent F:** *"Model documentation is very important. It should include answers to: Why, what, who was involved, assumptions, model development, findings, results, etc.)".*
- **Respondent G:** *"Data are not usually available on time, and they are usually in the wrong format".*
- **Respondent J:** *"Simulation time is not essential. Actual computers are very fast".*

4.3.3 Section 3: Results obtained from simulation models

A common use of simulation models includes obtaining quantitative results in order to take further actions or proceed to the implementation stage. This section of the survey aims to collect information on the importance that most common performance measures have to practitioners. However, this information is insufficient to provide answers as to the way in which these performance measures are used in practice. In some cases, effective measures (relative importance, ranking, etc.) might be sufficient to select or discard the option in consideration, while in other cases, an accurate result is necessary in order to decide, for example, if the model in question is a valid representation of the real system. This question is addressed in the second part of this section.

4.3.3.1 Variety of results

Table 23 and Figure 35 summarise the 'closed' responses and accumulative weight given by the respondents in this sub-section of the questionnaire. Some quotes from respondents are given below:

DESCRIPTION	RATES													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Rate	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Utilisation	●	●	●	●	○	⦿	⦿	●	⦿	●	●	⦿	⦿	N
Contribution	○	○	○	○	○	○	○	○	○	○	○	○	○	N
Throughput	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Lead time	⦿	○	⦿	○	⦿	●	⦿	⦿	○	○	⦿	⦿	○	N
Work in progress	●	●	●	●	●	●	●	⦿	⦿	⦿	⦿	●	●	⦿
Waiting time	⦿	○	⦿	○	⦿	○	○	○	○	○	○	○	○	N
Note: ● = Essential; ⦿ = Important; ◎ = Avoidable; N = Not answered.														

Table 23: Variety of results and their importance

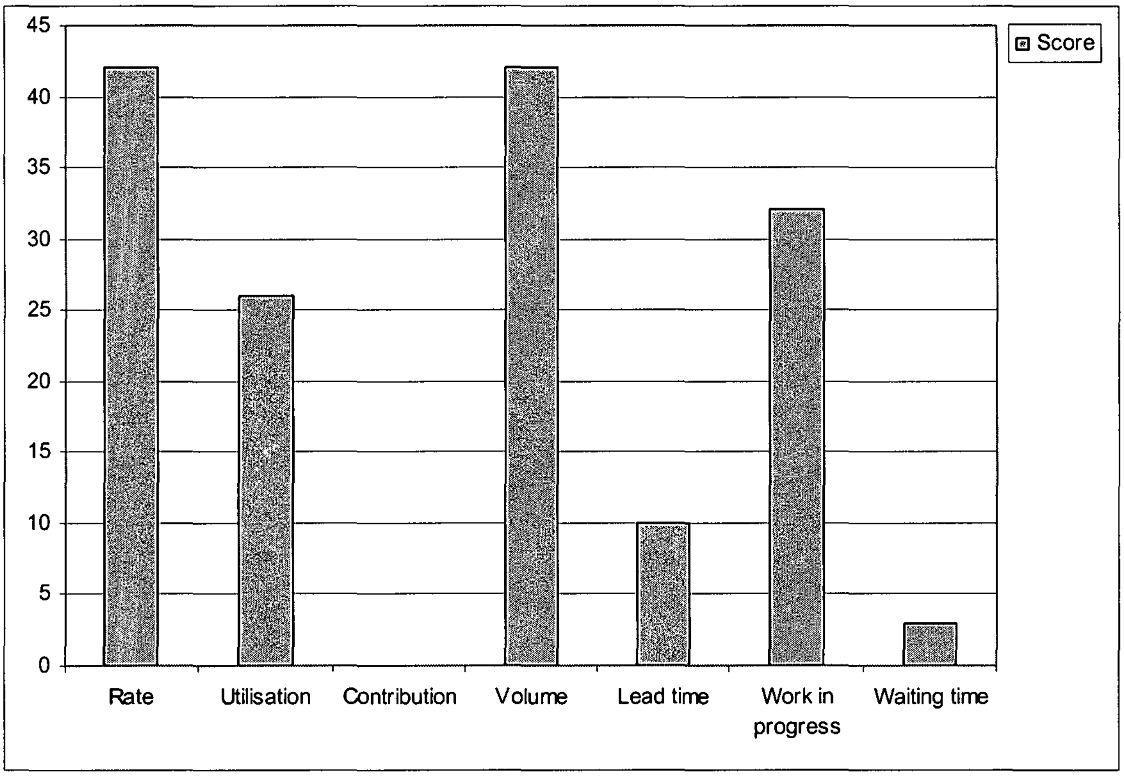


Figure 35: Accumulative importance of variety of results

- **Respondent C:** *"Utilisation and throughput are the most important measures. Work in progress is also important".*
- **Respondent F:** *"Rates, utilisation and throughput is a must!".*

4.3.3.2 Quality of results

Table 24 and Figure 36 summarise the 'closed' responses and accumulative weight given by the respondents in this sub-section of the questionnaire. Some quotes from respondents are given below:

DESCRIPTION	RATES													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Effective	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Accurate	●	⊙	●	●	●	⊙	⊙	○	●	⊙	●	⊙	⊙	⊙

Note: ● = Essential; ⊙ = Important; ○ = Avoidable; N = Not answered.

Table 24: Quality of results and their importance

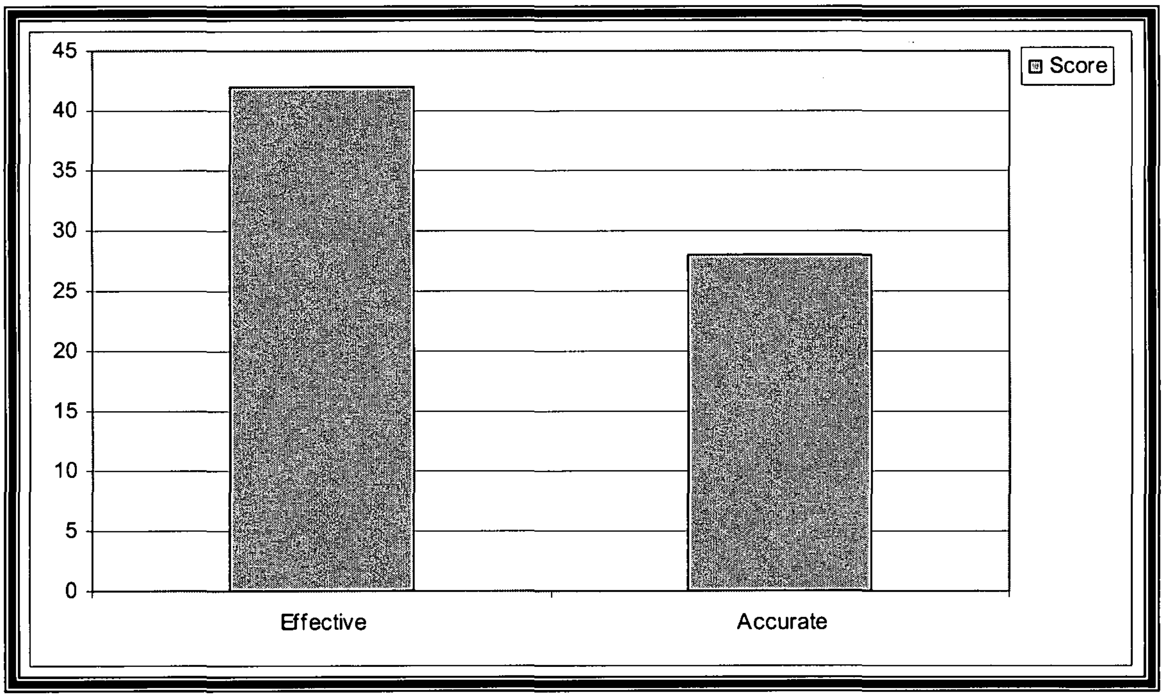


Figure 36: Accumulative importance of quality of results

- **Respondent A:** *"We target 5% accuracy in our models".*
- **Respondent B:** *"With ARENA I obtained an accuracy of 99.98% in a new plant development for a simulation of 16 weeks".*
- **Respondent D:** *"Producing 3600 cars, usual error is about 14 cars (0.4%)".*

- **Respondent F:** *"Accuracy must be more precise. The target must be optimisation".*
- **Respondent H:** *"A SD tool does not require much accuracy. It is enough to focus on decision making (e.g. inform the user if a new proposed solution is better/worst than the previous one)".*
- **Respondent L:** *"The tool should support decisions, thus, it is absolutely essential that the tool provides numeric outputs".*

4.3.4 Section 4: Utilisation of simulation models

This section is linked to the previous one. While section 3 of the questionnaire aimed to obtain the most valuable performance measures, this section aims to gather information on how the practitioners analyse the models and their use of graphical and statistical aids. Results obtained from this section will enable the researcher to build the features related with the model interaction and display of results.

4.3.4.1 Analysis of models

Table 25 and Figure 37 summarise the 'closed' responses and accumulative weight given by the respondents in this sub-section of the questionnaire. Some quotes from respondents are given below:

DESCRIPTION	RATES													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Step by step	⦿	⦿	●	●	⦿	⦿	○	○	●	●	○	○	⦿	N
Track parts	○	●	⦿	⦿	⦿	N	○	○	○	○	⦿	○	○	N
Animations in 'real time'	⦿	⦿	⦿	N	○	⦿	⦿	●	⦿	⦿	⦿	○	⦿	⦿
Variable simulation speed	⦿	●	●	⦿	⦿	⦿	○	○	⦿	⦿	○	○	○	N
Modification of the resolution of the system	○	○	○	N	○	●	●	⦿	⦿	○	○	●	●	⦿
Statistics	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Graphics	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Reports	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Note: ● = Essential; ⦿ = Important; ○ = Avoidable; N = Not answered.

Table 25: Types of models' analysis and their importance

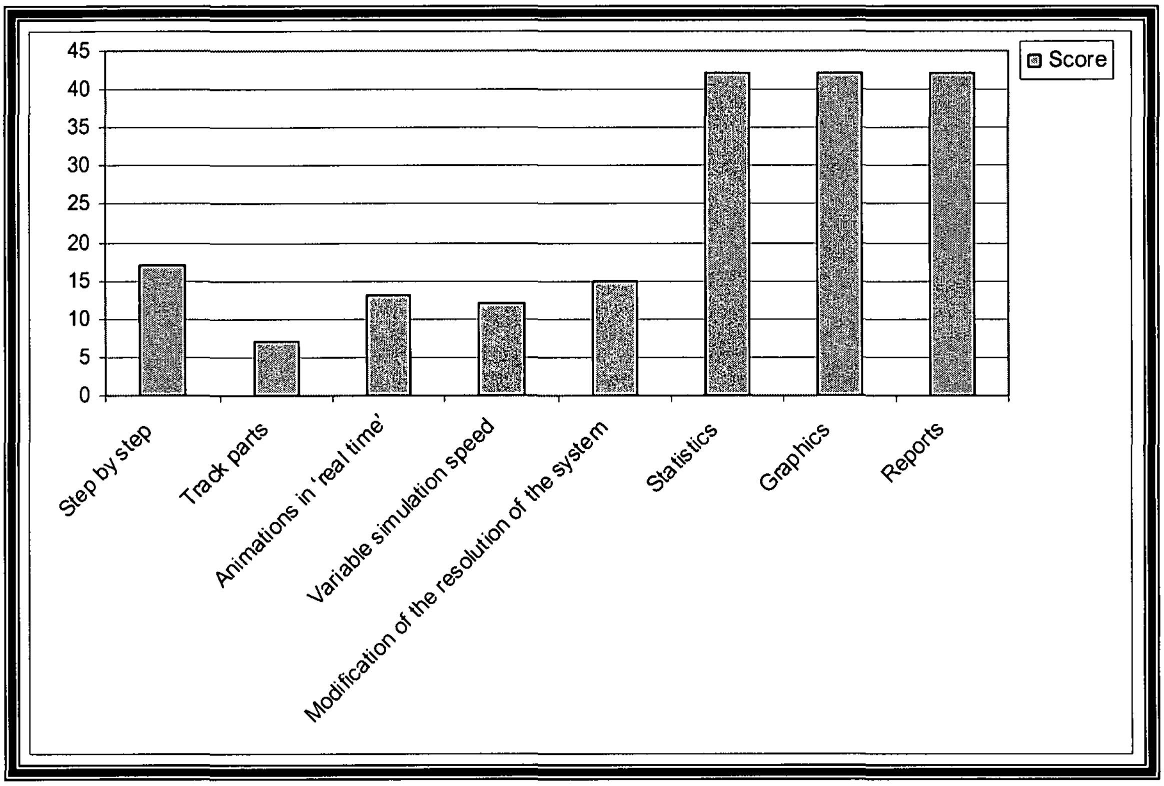


Figure 37: Accumulative importance of different types of model's analysis

- **Respondent A:** *"Step-by-step analysis is very important, especially to debug large systems".*
- **Respondent B:** *"Animations increase the confidence. If it is aided with 'part tracking' the user can see how the system behaves".*
- **Respondent C:** *"Hide any feature that allows the user to change the resolution of the model. If not, it will become another parameter to justify solutions!".*
- **Respondent F:** *"Actual SD graphics are really bad".*
- **Respondent I:** *"Dynamic graphs and long term reports help to understand the system".*
- **Respondent J:** *"I only use step-by-step simulation the first time I run the system to improve my confidence. Then, it is not essential".*
- **Respondent H:** *"When simulating flows, the SD tool should smooth the graphics to avoid oscillations".*
- **Respondent M:** *"I run the model without animations. The quicker the simulation the better it is".*

4.3.5 Section 5: Resources involved through the modelling process

The final section of this survey is aimed at gathering information on the resources (tangibles and non-tangibles) that practitioners require when carrying out a simulation project. As was stated in Section 2.3.3, the evolution of the simulation technique has always been closely related to computers. In addition, Section 2.3.3 highlighted the importance of the tools (in terms of interface and features) to the successful exploitation of simulation in industry. Thus, this section begins by asking the participants about the tool they use the most to simulate. This question is followed by an open discussion where participants can express their views about the requirements they consider most relevant.

PARTICIPANT	SOFTWARE
A	Simple ++
B	Arena
C	Witness
D	Witness
E	Witness
F	Extend
G	Witness
H	Witness
I	Arena
J	Arena
K	Witness
L	Witness
M	Witness
N	ProModel

Table 26: Simulation used by practitioners

- **Respondent A:** *"A new tool could be useful if it requires as few skills as possible".*
- **Respondent D:** *"Computer performance is still not enough; models take too long to run".*
- **Respondent F:** *"Skills depend on the context in which simulation is used".*
- **Respondent H:** *"It is important that users know the limitation of the tool beforehand".*

- **Respondent J:** *"The most important resource is the data. Most of the time, it is difficult to obtain".*
- **Respondent M:** *"Good analytical skills are important. Even the most easy to use software is not useful if the user cannot interpret the results".*
- **Respondent N:** *"The new tool should simulate the models at least as fast as the actual ones".*

4.3.6 General comments

In some cases, the questionnaire was followed by a small debate about the opportunities and characteristics of this research. Participants made valuable contributions about concerns or ideas that could help the development of the research. This section includes comments made by respondents during this stage.

- **Respondent A:** *"Try to make 'blocks' like DES tools to simplify the simulation task".*
- **Respondent B:** *"DES can do everything better than SD".*
- **Respondent D:** *"Focus the attention on creating a good front-end, but also to be customisable".*
- **Respondent E:** *"Although you might build a better tool, people will still use DES, because it does everything".*
- **Respondent F:** *"In order to define the specifications, it is very important to analyse where the tool is going to fit. Who the targeted customers are and its applications".*
- **Respondent F:** *"Consider building a 'hybrid' tool that takes the best of each technique (continuous and discrete)".*
- **Respondent F:** *"SD is better for unstructured problems".*
- **Respondent G:** *"If the tool aims simulation at a higher level, it is convenient to lay out a clear display, easy and intuitive".*
- **Respondent H:** *"A SD modelling tool could be useful if it is focussed to deal with rapid prototypes".*
- **Respondent H:** *"'Drag' and 'drop' is very important when using the tool. People like the mouse".*

- **Respondent K:** *"The software should be applicable in different scenarios; not just in manufacturing ones".*

4.4 Key findings

The results of the data analysis and discussion are presented here. The analysis is structured in five sections, each one corresponding to a survey section. The analysis, in addition to the literature, will be used in next chapter to define the specifications of the tool. It will also be used to fulfil the first objective of this research, related to understanding the business needs.

4.4.1 Section 1: What do practitioners simulate / what a manufacturing simulator based on SD should simulate

4.4.1.1 Model elements

Literature established three basic elements that most DES models include, namely: (i) Parts, (ii) Buffers and (iii) Machines or activities (Section 2.2.1), although actual DES tools include many more in order to facilitate the modelling task. SD based tools, however, are primarily based on three main elements, namely: (i) Rates, (ii) Levels, and (iii) Flows of material/information.

When practitioners were asked about the model's elements that they considered to be important/essential, they ratified this concept; 100% of respondents considered essential the inclusion of 'Parts', 'Buffers' and 'Machines' within a simulation tool. Thus, it appears apparent that the lack of any of these elements in a simulation tool is likely to reduce its usefulness, and could even make the tool ineffective.

Different answers were provided when practitioners were asked about how important/essential other elements are when constructing a model. Although practitioners usually use elements such as 'Labour' or 'Conveyors', alternatives were provided to model some of these elements without the development of specific 'objects' for them, in order to reduce the complexity of the new tool. Thus, suggestions such as *"Labour and conveyor can be simulated using machines (Respondent A)"* were given.

The 'closed' questionnaire was followed by an 'open' discussion about the elements described above. Some practitioners expressed their views about the 'continuous' treatment that the SD technique gives to the elements of the model. Thus, respondents C, E and I seemed to dislike the display of parts through the system using 'real' (e.g.

4.97) numbers, and suggested that 'integer' (e.g. 5) numbers should be used instead. Comments such as "*Materials flows seem confusing. I prefer the 'discrete factory'* (Respondent I)" were given to justify their views. This issue will be analysed further in Chapter 5 when creating the specifications of the tool.

4.4.1.2 Models flows

Literature revealed that complexity of manufacturing systems has grown during the last few decades (Section 2.3.1). Responses provided in this section aimed to determine the type of flows (connections between elements) simulated in industry. This does not imply the rules that govern the elements of the model, but only the alternative paths that exist within it. Responses were unanimous when dealing with the 'connections' between different elements of the model. 100% of the respondents stated that 'N to N' links are required in industry, where 'N to N' means that 'more than one machine' can be connected to 'more than one machine'. An example of this can be a model consisting of four machines (A, B, C, D) where they are all inter-linked (A-B, A-C, A-D, B-C, B-D, C-D). A comment made by respondent 'A' reinforces this position: *"If the tool does not include N to N flows, it is non-useful in industry"*

Special mention must be given to 'batches', because it is a feature demanded in industry (rated as essential by four respondents). Respondent C viewed mathematical continuous models to be less suited than discrete systems when dealing with batches and stated: *"Materials might come in batches, but a SD model will treat them as levels"*. However, in some cases, prioritisation of elements can be made to emulate the behaviour of batches.

4.4.1.3 Model's accessory elements

The response chart displays 'constants' and 'variables' as key features to include in the simulator to develop a valuable tool for industry (ratified by 100% of respondents). Practitioners extensively use random numbers when dealing with real cases (Section 2.2); stochastic in nature. However, their views were not homogeneous when asked about the importance of random numbers in a modelling tool aimed to simulate the systems at an aggregate level. While some practitioners still seemed to view random numbers as important (1) or even essential (4) elements of models, others seemed to find it avoidable (5) in this context. The number of respondents (4) that did not answer this question is significant.

Respondents also expressed the importance of breakdowns and setups when creating a model. However, respondents suggest that these can be simplified to average values if the tool is going to be used to simulate systems at an aggregated level. This simplification implies that a machine affected by any of these effects (breakdowns and/or setups) will in fact run slower, but will not have any stoppages. An interesting comment was made by Respondent 'A', who states that some analysis can not be made if the above mentioned simplification is made: *"If breakdowns are simplified, analysis like 'if this machine stops, how much time do I have?' cannot be done".*

4.4.1.4 Summary of key findings within this section of the questionnaire:

1. Elements to simulate 'parts', 'buffers' and 'machines' are essential.
2. Other elements such as 'labour' can sometimes be modelled by using the three elements described in (1).
3. 'Integer' numbers are sometimes preferred to 'real' ones.
4. 'N to N' link possibility is essential.
5. Variables that deal with information are required.
6. Setup times and breakdowns patterns can be simplified to simulate in a more aggregated level.

4.4.2 Section 2: How practitioners simulate / how should they simulate

Although literature highlights the importance of a structured approach when conducting a simulation project (Section 2.2), the low response obtained when practitioners were asked about the time they spend in each task of a simulation project is significant. A reason for this was pointed out by some respondents, who adduce to the high variability of this response, according the type of model in hand (detail vs. rough).

As can be appreciated from Table 22, data collection can consume up to 23% of the total time assigned to a simulation project (Respondent I). In addition, it is commonly agreed that it is a tedious task, accentuated by the incompatibility of data that often arises when trying to introduce collected data into the model (Respondent G): *"Data are not usually available on time, and they are usually in the wrong format".*

Model coding seems to be a reasonable task, appreciated by experienced practitioners. However, Respondent 'C' believed that if the tool is not going to be used by a specialist, it is necessary to emphasise the creation of an intuitive tool, or even allowing the user to introduce the data through a known environment (e.g. Microsoft Excel).

When modelling small systems, computer performance does not seem to be a problem (Respondent J): *"Simulation time is not essential. Actual computers are very fast"*, but there are some cases were larger models force the user to leave the computer running the model for days to obtain the results. If this time is multiplied by the number of experiments carried out with the model, the total time can be considerable. This view was expressed by Respondent 'B' in section 5 of this questionnaire.

4.4.2.1 Summary of key findings within this section of the questionnaire:

1. Time spent on data collection is usually high.
2. Computer performance can be a problem if the number of experiments is high or the model is very complex.

4.4.3 Section 3: Results obtained from simulation models

4.4.3.1 Variety of results

Section 2.3.2 and Section 2.2 revealed 'Productivity (or production rate)', 'Parts produced (or throughput), 'Capacity utilisation level', 'Delivery time (or lead time)', and 'Work in progress' as important performance measures for industry. The majority of practitioners validate this concept, by rating as 'essential' performance measures such as 'Rates' (100%); 'Throughput' (100%); 'Work in progress' (64.3%, the remainder of the participants rate it as important) and 'Utilisation' (50%, with 35.7% rating it as important). This view is summarised by Respondent C, who stated: *"Utilisation and throughput are the most important measures. Work in progress is also important"*.

'Lead time' is also considered to be an important performance measure. Although only one respondent rated it as 'essential', seven others considered this performance measure as important. However, due to the 'continuous' nature of SD based methods, individual parts can not be identified and thus, this measure might be difficult to obtain in practice.

4.4.3.2 Quality of results

Depending upon the type of decision that needs to be taken, different scales of measures might be required. For example, if a decision has to be made in order to implement a particular design out of a given number, relative performance measures might be sufficient. Thus, the particular set that obtains the highest score will be the most appropriate for implementation. This type of result can be considered as 'effective'. Conversely, some additional analysis might be required to obtain 'accurate' results. For example, if a new manufacturing plant is being designed to match a specific set of requirements, accurate values are needed to evaluate if the model will satisfy the set of specifications, or if further improvement is require in certain areas. This survey revealed that most practitioners do not find sufficient 'effective' results. Thus, 92.8% of the people interviewed considered it important (50%) or essential (42.8%) to provide the tool with algorithms that provide not only effective results but also accurate ones.

Special mention has to be given to the open discussion that followed the closed questionnaire of this section. Two respondents from two large automotive companies claimed to obtain their model's results with an accuracy of 99.98% and 99.6% respectively. Answers from these respondents are provided below:

- **Respondent B:** *"With ARENA, I obtained an accuracy of 99.98% in a new plant development for a simulation of 16 weeks".*
- **Respondent D:** *"Producing 3600 cars, usual error is about 14 cars (0.4%)".*

Although there is no evidence to challenge this claim, it does appear unrealistic that a simulation should provide this level of accuracy in these complex environments, since any model is by its nature an approximation of the real system. Other researchers suggest that in practice, models are typically only 80% accurate (Siebers, 2001). On this basis, it might be sufficient to interpret this response as a practitioner seeking as much accuracy as possible from a simulation model, rather than being overly concerned with the precise levels of accuracy offered.

4.4.3.3 Summary of key findings within this section of the questionnaire:

1. Machine rates, product throughput, work in progress and utilisation are the key performance measures.
2. Can also be useful to include the possibility of obtaining 'lead time' measures.
3. Effective measures are not enough; they also need to be accurate.

4.4.4 Section 4: Utilisation of simulation models

Literature often relates simulation to statistical analysis. Rubinstein and Melamed (1998) state that *"the simulation approach merely generates possible histories and then calculates statistics from them".* Practitioners also closely related simulation with statistical analysis, and all of them found statistical results an essential feature to include in a simulation tool. In addition, all interviewees found graphics and reports were essential for representing results. While graphics such as 'time series' or 'histograms' were found useful to represent, for example, productivity or machines cycle times, the importance of reports was also highlighted in order to carry out a detailed analysis of the system. This view is shared by Respondent I who stated: *"Dynamic graphs and long term reports helps the understanding of the system"*

This section of the questionnaire also aimed to gather information about the type of analysis that practitioners carry out with the models. 'Step by step' analysis was found to be essential for 28.6% of the respondents and important for another 35.7%. This type of analysis was mainly used to search for and eliminate malfunctioning elements or errors, or improve the modeller's confidence in the model. *"Step-by-step analysis is very important, especially to debug large systems* (Respondent A)*". "I only use step-by-step simulation first time I run the system to improve my confidence* (Respondent J)*".* On the contrary, once the system has been verified and confidence in the model exists, practitioners appear to disable this feature because it slows down the simulation. *"I run the model without animations. The quicker the simulation the better it is* (Respondent M)*".* Similarly, some practitioners use animations in 'real time' (71.4% rate it as important and 7.1% as essential), or varying the simulation speed (42.8% rate it as important and 14.3% as essential) because it helps in understanding the system.

Modification of the resolution of the system offered divergent opinions that depended on the business sector of the participants. While all members from academia (Respondents F, G, L and M) found essential the inclusion of an element that allows the user to modify the time interval between two consecutives stoppages, or 'delta time' (Section 2.2.2.4); inherent characteristics of SD based models, only one industrial participant out of eight (Respondent I) found this feature important. Moreover, some respondents expressed their disagreement about the inclusion of this feature in the simulator. Thus, Respondent 'C' stated: *"Hide any feature that allows the user to change the resolution of the model. If not, it will become another parameter to justify solutions!".*

4.4.4.1 Summary of key findings within this section of the questionnaire:

1. Statistical analysis is essential. It can be provided in 'time series' or 'histograms'.
2. Real time simulation is appreciated by industry, which also finds features such as 'step by step analysis' useful.
3. Contradictions exist about the inclusion of the Delta Time (DT) concept in the tool. While academia finds it essential, industrialists seem to dislike it.

4.4.5 Section 5: Resources involved thorough the modelling process

The final section of this survey aimed to gather information on the resources that practitioners require when carrying out a simulation project. Extensive literature exists about the different types of simulations and a number of guidelines are provided for selecting an appropriate simulation tool (Hlupic and Paul, 1996; Hlupic, 2000) (see Section 2.4.3). Within the population of this survey, Witness (Lanner, 2003) was the most popular tool, being used by 57.1% of the interviewees, followed by Arena (Rockwell Software, 2003), which was used by 21.4% of respondents. The reasons offered for using these particular tools include: (i) Features of the software, (ii) Facility of use, (iii) Standardisation within the organisation and (iv) Previous experience with this tool.

The evolution of simulation computer tools and, more specifically, their graphical interfaces, has expanded the range of users for this method of analysis. It is also commonly agreed that actual simulation tools are now easier to use and more intuitive than when they were originally developed. However, practitioners considered simulation as an area where skills are still required. Respondent 'M' agrees with this concept, stating that: *"Good analytical skills are important. Even the most easy to use software is not useful if the user cannot interpret the results"*. In addition, the actual tools included a considerable number of features, and each new version usually includes new features. Thus, continuous training is required by practitioners to keep their skills up-to-date.

4.4.5.1 Summary of key findings within this section of the questionnaire:

1. Witness, from Lanner Group is a popular tool within both industry and academia.
2. Although graphical capabilities improve the usability of the tool, analytical skills are required in order to, for example, interpret the results.

4.4.6 General comments

All the participants of this survey were aware of the DES technique and had used tools based on this technique. However, although the knowledge of SD was obvious for academic participants, not all industrialists were aware of the peculiarities of this technique. This is supported by the fact that none of the industrialists interviewed had recently used any SD based tool. Moreover, if the user interface is similar to the actual DES based tools (based in iconic representation), participants suggested that a new tool based on SD principles would be easier to introduce into industry.

In addition, there are some comments from industrial participants that discourage the use of a tool based on the SD principles within these types of user. Comments such as: *"DES can do everything better than SD* (Respondent B)*"* or *"Although you build a better tool, people will still use DES, because it does everything* (Respondent E)*"* indicate that the tool based on SD principles might be better suited for users with less expertise that require analysis with higher levels of aggregation.

Finally, participants expressed their views about what a good front-end should look like. Simulation users recognised the value of a graphical front-end, preferably customisable (Respondent D). If the tool is going to be used by non experts, easy and intuitive displays are also appreciated (Respondent G). Finally, the use of the mouse (with 'drag' and 'drop' characteristics) was suggested as a desirable feature (Respondent H).

4.5 Summary

This chapter has detailed the design and content of the interviews with respondents through the identification of aspects such as: (i) 'What' and 'how' practitioners simulate, (ii) Results obtained from the tools and their analysis and (iii) Resources involved. The second part of this chapter compiles and rates the responses obtained from the 'closed' questionnaire used within the survey. It also provides the key comments collected within the 'open' discussion.

Finally, the data obtained from the interviews is presented. The analysis of this data has focused on answering the first objective of the research aim: "*Understand what is needed of a SD modelling tool for it to complement the needs of a manufacturing system designer*".

The quantitative analysis carried out for the 'closed' questions of the survey provided some clear examples of features that need to be included in the tool in order to make it more suitable for industry. Examples of these are 'machines', 'buffers', 'statistical analysis', 'productivity measures', etc. However, some results need further analysis before deciding on their inclusion within the tool. Examples of these are the treatment of the discrete variables, such as number of products (while SD treats them as a continuous value, industrialists seemed to prefer them as discrete) or how performance measures such as 'lead time' can be obtained. In addition, the 'open' discussion carried out during the interview has also provided useful information that helped to increase the acceptability of the tool within industry.

This chapter has increased the understanding of the application of simulation in industry, and its results will be used in the next chapter to detail the final specifications of the tool. The interviews have also provided an insight into where a new tool could fit in industry.

Chapter 5

DESIGNING AND DEVELOPING A MANUFACTURING MODELLING TOOL BASED ON SYSTEM DYNAMICS PRINCIPLES

Chapter 4 described the first stage of the research, gathering information about the application of simulation in industry in order to validate and complement the information obtained from literature, and to classify the importance of end-user needs. This chapter addresses the second research objective, through the design and development of a modelling tool based on the core System Dynamics capabilities and tailored to manufacturing system design. The structure of this chapter is as follows:

1. Methodology for developing a modelling tool
2. Needs analysis, concept definition and requirements specification
3. Designing the interactive and analytical capacity of the tool
4. Implementation of the tool, integration and tests
5. Software operation

5.1 Methodology for developing a modelling tool

Literature has shown that the tools used within the two different simulation techniques, SD and DES, both have graphical interfaces (see Section 2.4.3). However, the diagramming conventions used within these two methodologies are different, and the iconic representation of DES based tools appears to be preferred by practitioners. The objective of this stage of the research is to design and develop a modelling tool based on the SD principles but using the iconic representation that actual DES tools have, in order to truly compare the characteristics of SD and DES in industry.

Software development is *"like house-building. Before you start actually making something, you have to think through carefully what it is you want, and have a design to work from"* (Meek *et al.*, 1983). Thus, much of the success in software engineering is related to the method used to design and develop the tool (Zelkowitz *et al.*, 1979). The reasons for this include the difficulty of viewing in advance the complexity of large projects due to the interdisciplinary nature of most common software projects. Thus, in order to improve control over the development of the project, software managers have identified several steps through which software projects pass; these steps are collectively called the 'software development life cycle' (Zelkowitz *et al.*, 1979) or the 'waterfall life span' (Royce, 1998; MacKenzie *et al.*, 2002). Although other approaches exist, the *"waterfall life span is perhaps the most widely known and most idealised form of a lifecycle"* (Cugola and Ghezzi, 1998). An overview of this method, chosen for designing and developing the tool was presented in Section 3.5.3. The main steps of 'the waterfall life span', including the verification and validation links, are shown in Figure 38. In this context, verification is concern with 'building the thing right', guaranteeing that the artefacts produced in a particular phase conform to specifications and requirements established in the preceding phase(s). Conversely, validation refers to 'building the right thing', guaranteeing that the software product actually meets user needs and expectations (MacKenzie *et al.* 2002).

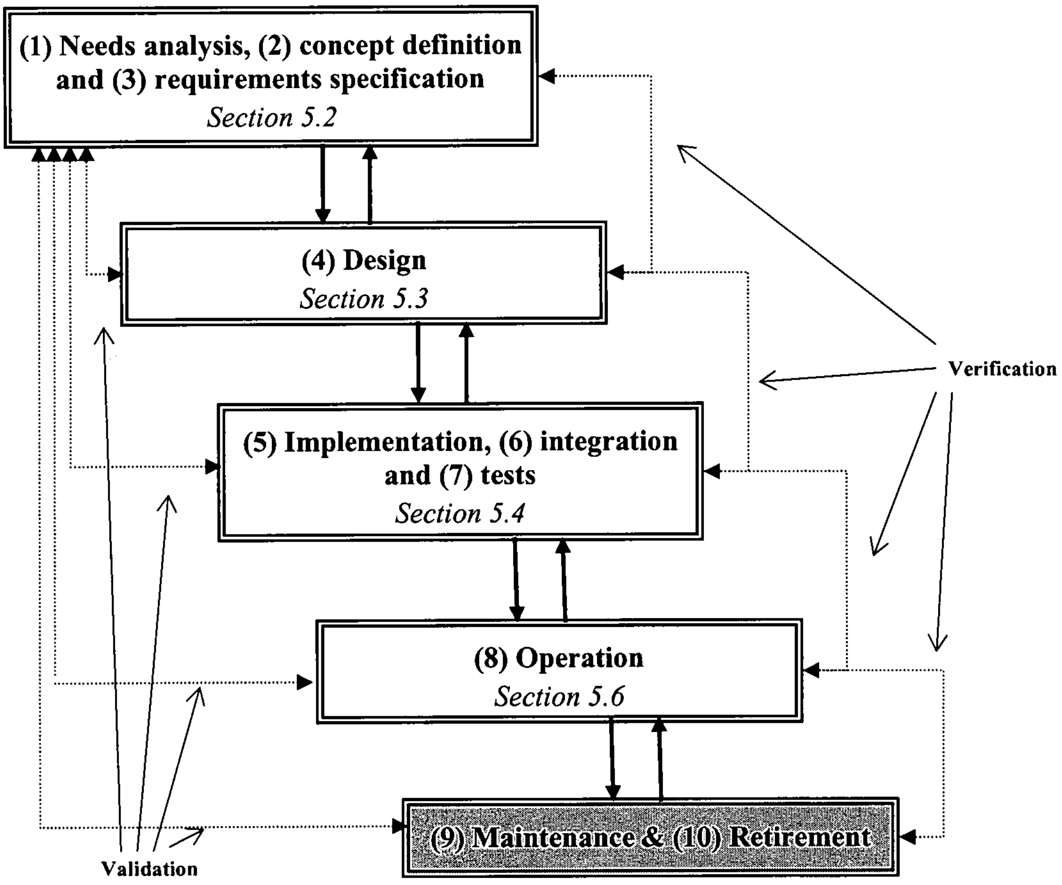


Figure 38: The waterfall life span (after Royce, 1998)

It is important to note here that 'maintenance' and 'retirement' stages are not included in this research because the tool is developed with prototype purposes.

The waterfall life span process can easily be explained with an example (see Figure 39). It starts with the perception of a need in the real world. When explicitly stated, these needs represent the requirements. However, the computer cannot solve the problem directly, and a model of the problem is required to represent the specification, determining how the process is to occur. This is represented in the design phase. Since the programme must be used to solve the real-world problem, the conversion of this abstract design into an executing programme represents the implementation stages (coding and testing). Finally, the maintenance stage closes the loop to altered requirements, etc.

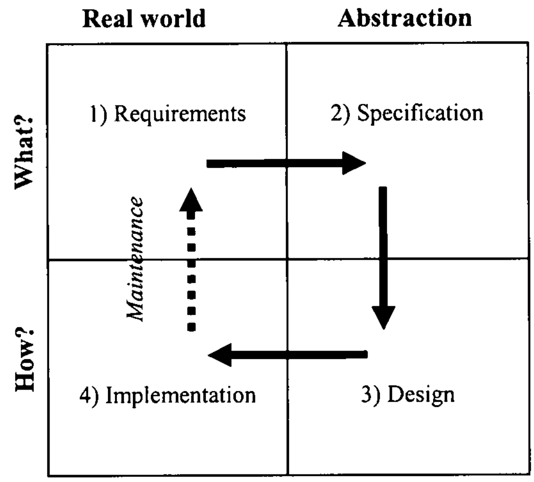


Figure 39: Software life cycle progression (Zelkowitz et al., 1979)

5.1.1 Needs analysis, concept definition and requirements specification

This first stage of software design has many similarities with the first stage of the method used to apply simulation in industry (described in Section 2.4.1). It defines the requirements for an acceptable solution to the problem. In addition, the concepts defined in this stage can help in identifying the user's preferences and conflicting constraints (Heckel, 1991). They are also used through all stages for validating if the developed features match the industrial needs (see Figure 38). Requirements specification, however, seeks to define precisely 'what' the software is to do (Zelkowitz *et al.*, 1979) although without specifying 'how' to do it (Easteal and Davies, 1989). It is important to appreciate that the specification of requirements fulfils a dual role. On the one hand, it represents a form of contract between the customer and the analyst. On the other hand, it represents the starting point for the design phase and must therefore be easily interpreted by the software analyst (Easteal and Davies, 1989). The more precise the specifications, the less likely that errors will occur in a software development project. Literature is commonly used for background reading in the area of the problem, both to gain a greater understanding and to find clues towards methods of solution (Meek *et al.*, 1983). In addition, it is advisable to collect the usability preferences of the end-user (who might be the customer or a third person) to evaluate the customisation of the software to its preferences, and therefore, reduce its learning curve.

Considering the reasons given above, in this research, the analysis needs and later requirements specifications have been developed on the basis of the following inputs (illustrated in Figure 40): (i) Simulation literature review, (ii) Application of simulation in industry, (iii) Industrial needs, (iv) Scope of this research. In addition, existing simulation tools have also been analysed in order to build a user interface similar to existent tools.

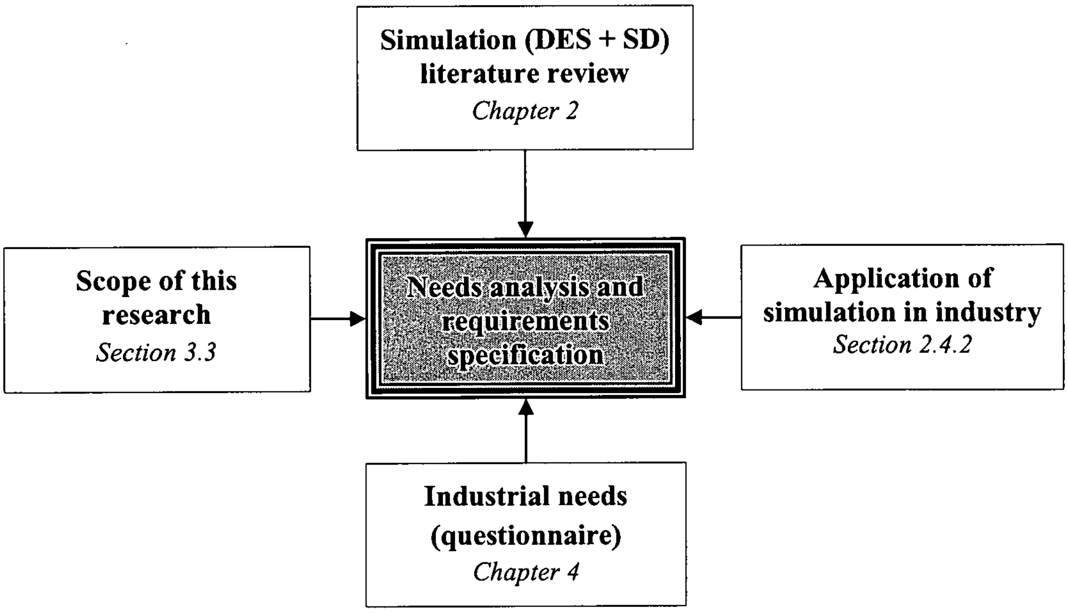


Figure 40: Inputs used to define software's needs and requirements

5.1.2 Design of the tool

With the commencement of the design stage, the attention of the software developers focuses on the question of how the user's requirements are to be implemented (Easteal and Davies, 1989). In this stage, the algorithms and user interfaces defined in the specification stage are developed, and the overall structure of the computer system takes shape (Zelkowitz *et al.*, 1979). User-friendliness or easy of use usually refers to factors such as graphics, input and output (Klemola and Turunen, 2001). It is commonly agreed that graphical user interfaces aid the user in the modelling task and expand the range of final users (see Section 2.3.3), although one disadvantage with easy-to-use graphical user interfaces is that they can be used easily without sufficient knowledge of what the calculations are based on. However, most engineering companies and industrial companies prefer easy-to-use programmes (Heckel, 1991; Mayhew, 1992; Klemola and Turunen, 2001). An exception to this rule occurs when the software is designed to be used by experts, where primitive user interfaces can sometimes be preferred or even more efficient and, therefore, the investment required to build graphical interfaces is not justified (Mayhew, 1992; Klemola and Turunen, 2001).

Several approaches for software design exist. For example, Easteal and Davies (1989) cite two similar design techniques, named 'structural design' (developed by Yourdon and Constantine, 1979) and 'composite design' (developed by Myers, 1975) and recommend their use due to their simplicity. These methodologies are also cited by Zelkowitz *et al.* (1979), who also describe other design strategies, namely:

1. Problem Statement Language (PSL/PSA), language designed to express functional and performance requirements.
2. Structured Analysis and Design Technique (SADT). Its language is based upon a hierarchically structured set of diagrams, each box in the diagram being defined in greater detail by another diagram.
3. Software Requirements Engineering Methodology (SREM), based on the PSL methodology.

A 'structured design' approach is recommended for software design, because it gradually reduces the complexity (Yourdon and Constantine, 1979; cited in Easteal and Davies, 1989). This approach, also followed by Easteal and Davies (1989), has been chosen as adequate for the complexity level and purpose of this research. It consists of three distinct activities which take place in sequence, namely: (i) Initial or overall design, (ii) Detailed design, and (iii) Data structure design.

5.1.2.1 Initial or overall design

Initial design is concerned with establishing the overall 'shape' of the software by dividing it into basic blocks and determining the ways in which they should interface with each other. This stage of the design process helps to decide, for example, if a Multi Document Interface (MDI) or a Single Document Interface (SDI) is more appropriate. These two types of interfaces are discussed in detail by Petroutsos and Hough, (1999). Basically, a MDI consists in a main (also called parent) form, which contains the rest of the forms (also called child) that the user is going to interact with. Conversely, SDI consists of a unique or several independent forms. Petroutsos and Hough (1999) recommend MDI for those applications that contain several inter-related forms, because it provides a clear environment by incorporating the common functions and buttons in a main form, and leaves the specific functionality to the child forms. This view appears to be supported by simulation software developers, since all the tools analysed in Section 2.4.3 (Witness, Stella/iThink, etc.) are designed using MDIs.

Thus, a graphical front-end based on a MDI has been chosen for the construction of the tool. The child forms required in this MDI environment are described in Section 5.3.

5.1.2.2 Detailed design

Detailed design involves taking each form and defining the processes that should take place therein; and also the chosen elements or components to implement it. Nowadays, the *"number of graphical components available in modern programming languages are endless"* (Petroutsos, 2000). In addition to the standard components provided by the programming language, a large number of third-party developers offer enhanced or customised components to perform specific tasks. However, compatibility issues and the investment required to purchase (and sometimes, distribute) these components must be analysed.

Software developers such as Zelkowitz *et al.*, (1979); Heckel, (1991); Mayhew (1992); Mullet and Sano (1995) and Garrick (1999), as well as simulation experts such as Pidd (1988) and Sterman (2000) provide useful guidelines and/or ideas for developing user interfaces.

A summary of the guidelines provided by these authors is as follows:

- Use system tools where they are available. Using established tools for things such as common dialogues (new, open, save, etc.) provides users with an advantage of already knowing how to use parts of the applications before ever running it.

- Follow established conventions for the layout of interface elements. This provides a professional appearance to the application and offers the users an extra degree of familiarity with the application.
- Structure the user's interface in an efficient, clear and predictable way. The adequate use of menus, buttons bar, lists, etc. can reduce the complexity of the final interface.
- Communicate visually; avoid frustrating the user. Make your design simple, but not too simple.
- Iconic displays are perhaps the most obvious way to display the simulation graphically.
- Chart displays help the user to understand the results. They can take the form of histograms, line graphs and bar charts.
- Know your subject and audience; speak the user's language.
- Put control in the hands of the users. Allow the users to customise the applications, and therefore, increase their satisfaction and productivity.
- Do not assume that your application is the only programme in use. Do not attempt to control settings that are normally handled by the user.
- Integrate the application with the operating system.

5.1.2.3 Data structure design

The final stage is concerned with choosing the data structures that will be involved in the processing. Before tackling data design proper, it is necessary to remind the reader of its purpose; data has no intrinsic value. What value it has lies entirely in its ability to say something about the real world. Thus, data structure design will be divided by considering the manner in which the data are stored: (i) Internally (in the RAM memory) or (ii) Externally (in files).

Computer programmes usually require substantial data to run. For example, if the 'save' button is pressed, the system might check if the actual model has been already saved, or whether new modifications have to be stored. To store all the information required to run the system, different types of data are required. Easteal and Davies (1989) classify the data according to different patterns, such as: (i) Structures (array, record, sequence, tree), (ii) Type (integer, real, boolean, etc.) and (iii) Use (static, dynamic, local, global, etc.). Section 5.4 explains the use of different types of data through the development stage.

Conversely, data are saved externally, for example, when it needs to be stored for future use, and the volatile memory of the computer is not an option. In addition, if the dynamic requirements of data are higher than the available memory of the computer, data has to be stored and accessed later from files to avoid system errors (Petroutsos and Hough, 1998). Thus, due to the uncertain size of the models and running horizon, the developed tool makes use of external files to store both model (elements properties, location, etc.) and running (performance measures, etc.) information.

There are a number of options to store data externally. According to Zelkowitz *et al.*, (1979) data can be stored in a: (i) Proprietary form or (ii) Standard form. In this context, 'proprietary' refers to data organised in a tailored way, whereas 'standard' refers to data stored following the structure of a popular format (for example, Excel or Access). Advantages of proprietary data include speed of reading/writing, due to the optimisation of the data structure for the problem in question (Zelkowitz *et al.*, 1979) and the possibility of encryption (Easteal and Davies, 1989). Conversely, the most important disadvantage of this type of data is the interaction with other software, which in many cases is impossible. Standard files, however, can be used with the 'source' programme and all other programmes that implement compatibility with that particular file type. Considering the reasoning given above, a 'standard' file type has been chosen to store external files; in particular, compatibility with Microsoft Excel is considered to be the most adequate because of its the popularity within industry, compatibility with modern simulation packages and both analytical and graphical capabilities of Excel.

5.1.3 Implementation of the tool

Implementation of the tool involves the electronic creation of the user interface and coding the functionality of the tool being developed. This can be done by: (i) Tailoring an existing tool, (ii) Developing the interface by using available modules and (iii) Writing the whole software from scratch (Meek *et al.*, 1983). Within the simulation context, three possibilities exist for the development of a tool:

1. Use of a commercial modelling tool based on the SD principles (for example, Stella/iThink) analysing the need for creating new modules or user interfaces.

2. Use of generic tools that allow modelling systems based on SD principles through specific modules (for example, MatLab with the Simulink module).
3. Use general purpose programming languages to write the code.

Commercial tools based on SD principles are those that can be used directly to model continuous systems. Many commercial tools based on the SD technique, such as Stella/iThink, Powersim, Vensim, Dynamo, etc exist in the marketplace. Including a wide variety of functions as standard, the model building tool can sometimes be personalised, allowing the user to model a variety of systems (Diaz, 2003). In addition, they minimise the number of errors, since less code is required. However, as stated in Section 2.2.2.3, the notation implemented within this type of tool can lack intuitiveness and offer a less user-friendly interface, amongst other reasons, due to the variety of uses they are designed for. Although modern SD based tools such as Powersim allow the user to develop user interfaces using a general purpose programming language, a full license (usually expensive) is needed in each computer that uses the developed interface in order to run the model (Powersim Software, 2003). Flexibility is an additional limitation of this software configuration; the interface must be designed to match exactly the way in which the source software (in this case, Powersim) manages the elements and use a compatible programming language. In addition, the communication between the developed user interface and the source programme makes the execution of the models slower (Diaz, 2003).

Some generic software such as Matlab (MathWorks, 2003), can support add-ons to benefit from the extended mathematics libraries included in the tool. These are useful tools for comparing the modelling of a system with different approaches, although their environments are usually based in a command line in which all the orders are introduced from the keyboard. They provide outputs in a graphical way, but the model is often built using equations and specific rules, rather than icons or another graphical representation.

Finally, some general purpose programming languages exist in the marketplace. They require programming skills and larger amounts of code, but the personalisation of the tool created is complete (Zelkowitz *et al.*, 1979). In addition, their object oriented (O-O) nature allows programmers to develop independent 'modules' or 'classes'. These modules can be designed to provide outputs to a given set of outputs, with a peculiarity that can be reused, and therefore, reduce coding time (Petroutsos and Hough, 1999).

A wide variety of programming languages exists, each one specialising in one specific area. The selection of a particular programming language to write the code is sometimes determined by either the preferences of the customer or the programmer (Zelkowitz *et* *al.*, 1979). However, in order to select an appropriate programming language for a particular project, some features can be compared (Petroutsos and Hough, 1999). The main features are: (i) Language level, (ii) Functionality, (iii) Expandability and (iv) Support.

5.1.3.1 Language level

Authors such as Brooks (1975) and Halstead (1977) (cited in Zelkowitz *et al.*, 1979) have mentioned that the number of lines of code produced by a programmer in a given time tends to be independent of the language used; thus higher level languages enhance productivity. High level languages, such as Visual Basic (Microsoft, 2003a) and Delphi (Borland, 2003), appears to be more productive than medium level languages such as C++ (Microsoft, 2003b) and low level languages such as Assembler (Microsoft, 2003c). However, high level languages enhance productivity at the expense of execution time, with each code line being slower to execute. However, with the increasing performance of computers, the trend is to: *"let the task be made easier for the programmer; let the computer do more work"* (Zelkowitz *et al.*, 1979).

5.1.3.2 Functionality

Although generic programming languages are designed to deal with a wide variety of scenarios, some specialisation does exist between the commercialised products. For example, Visual C++ is considered to be powerful when used to develop tools that must interact with other devices in real time (Huang *et al.*, 2002); FORTRAN (Microsoft, 2003e) is suited to the solution of many numerical problems in mathematics (Pidd, 1988), while the non-dependability of the platform in which it is used means that Java (Sun Microsystems, 2003) has found particular use on the World Wide Web (WWW).

Conversely, there are programming languages that allow the user to create easier graphical user interfaces, and to implement a powerful code. Examples of these programming languages are Delphi and Visual Basic. Delphi uses PASCAL as a programming language to build the code, making it more powerful in respect to Visual Basic because the PASCAL language was carefully designed to reduce inefficient code (Zelkowitz *et al.*, 1979). However, Delphi lacks in facility of use whereas Visual Basic allows the user to build graphical and user-friendly environments easily. Another advantage of Visual Basic is that it belongs to Microsoft; this makes its integration with Microsoft Office (Word, Excel and Access) (Microsoft, 2003d) easier, since Microsoft Office includes a reduced version of Visual Basic called 'Visual Basic for Applications (VBA)'.

In order to be executed by the computer, the code developed with any tool, needs to be translated. This is done by: (i) Compilers or (ii) Interpreters (Pidd, 1988).

- Compilers take all the source code developed and translates it into the executable code at one sweep. Thus, a single error will lead to an unsuccessful compilation. Once the compilation is successful, however, the resulting programme may be distributed and executed many times with no need for recompilation (Pidd, 1988).
- Interpreters take the source code one line at a time (or as small groups of lines) and translate this into machine code, being directly executed by the computer. Thus, if code errors do exist, they will only be detected if, during the execution of the programme, the execution of faulty code is attempted.

The use of one form of translation or the other is dependent on the tool used, rather than the preferences of the programmer. Thus, FORTRAN, PASCAL and C++ are languages that are always compiled, whereas Basic is usually interpreted but for which compilers are available (Pidd, 1988). This characteristic provides an advantage to Visual Basic, because the code can be interpreted to debug the programme easier (reducing the testing time) and compiled at the end in order to increase execution speed and portability.

5.1.3.3 Expandability and support

Today's high and medium level programming languages include large expandability possibilities by adding commercial modules (usually developed by third parties) (Petroutsos and Hough, 1999). Although many developers commercialise add-in modules that can be used in more than one programming language, it is still common to find add-ins that can only be used in the programming language that it is targeted toward (Petroutsos, 2000). In practice, both Visual Basic and Delphi are well supported because of the great number of users around the world, although Visual Basic is the chosen programming language at Cranfield University.

Table 27 and Table 28 summarise the main characteristics of the three different approaches for software development (use of commercial tools, generic tools or programming languages) and the features of the three main programming languages considered (C++, Delphi and Visual Basic).

Commercial tools based on SD	Generic tools with SD modelling add-ons	Programming languages
(+) Ready to use (-) Intuitiveness and environment	(+) Powerful libraries (-) Interface	(+) Personalisation and power (-) Programming skills

Table 27: Approaches for software development

Visual C++	Borland Delphi	Microsoft Visual Basic
(+) Execution speed is very high (-) Requires the most code to implement functions (-) Specially suited for real time applications (-) Not supported at Cranfield	(+) Large quantity of libraries and classes (add-in modules) (+) Execution speed is high (-) Difficulty to use (-) Requires more code than VB (-) Not supported at Cranfield	(+) Facility to use and debug, requires less code (+) Graphical capabilities (+) Integration with Microsoft Office (+) Large quantity of libraries and classes (-) Execution speed is slower

Table 28: Main features of C++, Delphi and Visual Basic.

Considering the strengths and weaknesses of the different approaches described in Table 27, the flexibility in terms of interface design, specifications, and no need of licenses to distribute the software means that programming languages have been selected as the most appropriate method for designing the tool. In particular, Visual Basic has been selected as the programming language, due to its facility to use, graphical capabilities and support at Cranfield.

5.1.4 Integration and tests

The testing stage can take up to half of the total effort (Zelkowitz *et al.*, 1979). Inadequately planned testing often results in rework of substantial pieces of the software. In addition, the tool has to be tested with representative data that allows the identification of a correct (or defective) functioning of the system (Zelkowitz *et al.*, 1979). Zelkowitz *et al.* (1979) divide the testing procedure into three distinct operations, namely: (i) Module testing, (ii) Integration testing and (iii) Systems testing.

- ‘Module testing’ involves the verification of each module with data supplied by the programmer.
- In the ‘integration testing’ stage, groups of components are tested together. This test usually identifies errors caused when linking different modules within a sub-system, or the whole system.

- 'System testing' causes a test of the completed system by an independent group. It is called a benchmark test if the performance of several systems is being compared.

When validating correct programme development, the term 'correct' can have many interpretations. Conway (1978) lists eight different meanings for a correct programme:

1. A programme contains no syntactic errors.
2. A programme contains no compilation errors or failures during programme execution.
3. There exists test data for which the programme gives correct answers.
4. For typical sets of test data, the programme gives correct answers.
5. For difficult sets of test data, the programme gives correct answers.
6. For all possible sets of data that are valid with respect to the problem specification, the programme gives correct answers.
7. For all possible sets of valid test data and all likely conditions of erroneous input, the programme gives correct answers.
8. For all possible input, the programme gives correct answers.

The achievement of each of these levels includes a cost for developing and testing the system. Thus, level '8' correctness is not always attainable or even needed. If the data are known to be correct, then level '6' might be sufficient. Also, if failures are sufficiently rare, the reliability of level '5' might be acceptable (Zelkowitz *et al.*, 1979).

The tool developed during this research has been extensively validated both as individual modules and as an integration level by the researcher. In the first case, all modules developed have been verified and validated with relevant data (when required) to ensure the correct functioning and the outputs provided by the module. In the second case, each module of the system has been probed individually to check the integration of their embedded modules and the communication between them. Finally, an example has been modelled with an audience. In addition, the system is used during the testing stage by a third party, who tested the system in different scenarios and manners.

5.2 Needs analysis, concept definition and requirements specification

As stated in Section 5.1.1, software requirements have been developed on the basis of: (i) Literature, (ii) Application of simulation in industry, (iii) Industrial needs, (iv) Existing simulation tools and (v) Scope of this research. These concepts are now expanded and lead into the main software requirements.

5.2.1 Simulation literature review

Section 2.1 and Section 2.2 have provided a literature review about the concepts of modelling, including simulation techniques, and the mechanisms of two forms of simulation (SD and DES). In addition, Section 2.4.2 has highlighted some differences between the two simulation approaches mentioned previously. Key inputs provided from this literature review, which will be used to develop the specifications of the tool, are compiled in Table 29.

Area	Input	Section
L1	Continuous manufacturing operate on product that is continually flowing.	2.1.4
L2	Simulation approaches calculate statistics from possible histories.	2.1.4
L3	SD is definitely not a highly refined and accurate tool.	2.1.4
L4	Hybrid simulations are often built using mathematical equations, and enhance by 'discrete-event' addition.	2.1.4
L5	Main DES elements are: (i) Parts, (ii) Buffers and (iii) Machines. They contain information attached in the form of attributes.	2.2.1
L6	SD models, in contrast to DES ones, are not closely related with stochastic numbers or discrete distributions.	2.2.1.1
L7	Activity cycle diagrams, causal loops diagrams and levels and rates diagrams tend to be confused when representing complex models.	2.2.1.2 2.2.2.3
L8	In SD, the stated variables change continuously with respect to time.	2.2.2
L9	SD models are basically based on levels, rates and converters.	2.2.2
L10	Quantitative SD involves differential equations that can be simplified using difference equations.	2.2.2.1
L11	SD models can also contain delays and non-linearities	2.2.2.1
L12	Feedbacks loops can be positive and negative	2.2.2.1
L13	The resolution of a SD simulation is determined by Delta Time (DT)	2.2.2.4
L14	DT decreases if the order level of loops increases to maintain accuracy	2.2.2.4

Table 29: Tool's requirements key inputs from literature review.

5.2.2 Application of simulation in industry

Section 2.3 and Section 2.4.1 have discussed the importance of simulation (including typical performance measures) and the main stages involved in a simulation project, respectively. Section 2.4.1 also emphasised the data collection and model building stages, due to the large amount of time required to conduct these stages. Key inputs provided from the process of application of simulation in industry are compiled in Table 30.

Area	Input	Section
A1	Main performance measures include: Scrap and rework, throughput / productivity, work in progress, labour cost, capacity utilisation, etc.	2.3.2
A2	Opportunity for cost reduction is higher in earlier stages of the design process.	2.3.2
A3	Graphical capabilities have increased the acceptance of simulation in industry.	2.3.3 2.4.3
A4	Cost and expertise needed prevent simulation from being more popular.	2.3.3 2.4.3
A5	Data collection and model development take up to 40% of the time.	2.4.1
A6	Empirical data are useful when the system does not change substantially.	2.4.2
A7	Model the minimum amount of detail required to achieve the project's objectives.	2.4.2.1
A8	Digital computation of continuous models is done transforming differential equations into difference equations.	2.4.2.2
A9	SD models consider variables as levels and that changes occur based on rates.	2.4.2.2
A10	Continuous models can be improved by incorporating discrete concepts.	2.4.2.2
A11	Integration of simulation tools with existing data are a concern.	2.4.2.3 2.4.3
A12	Most popular DES tool are Witness and Simul8	2.4.3

Table 30: Tool's requirements key inputs from the process of application of simulation.

5.2.3 Industrial needs

Industrial needs were obtained from the interviews conducted in Stage 1 of this research (see Section 4.3 and Section 4.4). These interviews were conducted with experts from different areas such as: (i) Academia, (ii) Practitioners and (iii) Software developers. Key inputs provided from the results obtained in this stage of the research are compiled in Table 31.

Area	Input	Section
Q1	Essential elements in simulation models are parts, buffers and machines	4.3.1.1
Q2	The tool needs to be able to model N to N flows in order to be useful	4.3.1.2
Q3	Constants and variables help to control the system's behaviour	4.3.1.3
Q4	Breakdowns and setups are also highly rated by respondents	4.3.1.3
Q5	Machine rates, product throughput, utilisation and work in progress are the most wanted performance measures	4.3.3.1
Q6	Effective measures are essential, but accuracy is also appreciated	4.3.3.2
Q7	Statistics, graphics and reports are needed in order to take actions	4.3.4.1
Q8	Data collection and model building are time consuming tasks	4.3.2.1

Table 31: Tool's requirements key inputs from the questionnaires.

5.2.4 Scope and limitations of this research

The development of a new simulator is both a complex and time consuming task. The purpose of its development within this research is justified in order to analyse if SD can be a true alternative to DES as a tool that suits manufacturing systems design practitioners. Thus, all features not related to the scope of this research (see Section 3.3) are discarded and will be presented as future work in Chapter 7. In addition, features such as 'step by step simulation', 'real time animation', etc. have also been discarded due to the considerable amount of time required to build such functionality into the tool.

5.2.5 Summary of tool's requirements

Considering the key inputs from literature, application of simulation in industry, practitioners' needs and scope of this research, the selected requirements for the tool are provided below.

5.2.5.1 System's elements and flows

Parts, buffers and machines were considered for implementation in the tool (L5, Q1). However, these elements are implemented using continuous elements (L1, L8, L9, A9), which means levels, rates and converters. Conversely, elements, such as labour, conveyors, shifts, vehicles, tracks, etc. have been discarded.

N to N flows (which include 1 to 1, 1 to N and N to 1) were also considered for implementation in the tool (Q2). When considering batches, SD treats materials as continuous flows. Thus, processing of batches is not straightforward in a pure SD tool and has been discarded. However, in some particular scenarios, batches can be simulated using the 'maximum quantity' attribute of parts.

Breakdowns and setup are considered useful by practitioners (Q4). These will be implemented as average numbers within the tool. Thus, distributions or real empirical data in the form of lists will not be implemented. Random numbers (L6) have also been discarded, due to the nature of continuous simulation. However, it can be implemented in the future to create a hybrid tool (L4, A10).

Delta Time (DT) is a key factor determining the resolution of the simulation (L13), thus, it is included in the tool. Literature stated that high order loops require a smaller DT to achieve similar accuracy (L14). Thus, considering this reason, and in order to simplify the tool, only first order linear control is implemented within the tool, and so effects produced by non-linearities are discarded.

5.2.5.2 Performance measures

Practitioners require results from simulation tools to take actions. Thus, the tool must provide numeric results. Accuracy of the results, however, will be in accordance with the average values introduced into the model (L3, Q6).

Performance measures included in the tool are: Machine rates, machine utilisation, throughput and work in progress (A1, Q5). However, other measures can sometimes be obtained using the output file provided by the tool (where all data obtained in the execution of the model is stored) and commercial spreadsheets such as Microsoft Excel.

In order to facilitate the analysis of the model (Q7), results obtained from the model include both reports and graphics. However, further graphics can be obtained by the procedure explained in the previous paragraph.

5.2.5.3 User interface

Diagramming conventions and graphical capabilities used in DES tools are more appreciated by practitioners (L7, A3). Thus, the tool emulates the iconic representation of popular DES tools.

In order to increase the compatibility of the data introduced/obtained to/from this tool, it is stored in a file compatible with Microsoft Excel and other products as Witness, etc. that can read/write these files (A5, A11, Q8). This is also done to expand the functionality of the tool by using a third company's software.

A summary of the requirements of the tool is provided in Table 32.

Area	Requirements
System elements	Part, buffer, machine (in a continuous manner)
	Breakdown and Setup (averages)
	Delta time
Flows	N to N, controlled by first order linear loops
Performance measures	Machine rate, utilisation, throughput, work in progress (numerically)
Results	Text format, graphics
User interface	Iconic representation
	Data stored in an Excel compatible file

Table 32: Requirements of the tool

5.3 Designing the interactive and analytical capacities of the tool

Requirements specification has defined what the tool is to do, but not how. It is in this design stage where the overall structure of the tool (explained in Section 5.3.1), as well as the embedded algorithm is designed (see Section 5.3.2).

5.3.1 User interface and interactive capacity of the tool

The literature revealed that, in the manufacturing system design context, simulation practitioners prefer an iconic representation. As stated by Pidd (1988), *"perhaps the most obvious way to display the simulation graphically is to use a set of icons to represent the entities... The idea of an iconic display is that the screen should resemble the simulated system in some recognisable way"*. Section 5.1.2.1 and Section 5.1.2.2 reasoned the adoption of a Multi Document Interface (MDI) and provided some guidelines for correct software design, respectively. To complement this information, an analysis of four popular simulation tools (2 SD and 2 DES) was conducted in order to identify the main similarities and differences in their user interfaces. The DES based tools selected to conduct this analysis were Witness and ProModel, while Stella/iThink and Powersim where the SD based tools selected. Examples of two models represented in Witness and Stella/iThink are shown in Figure 41 and Figure 42, respectively.

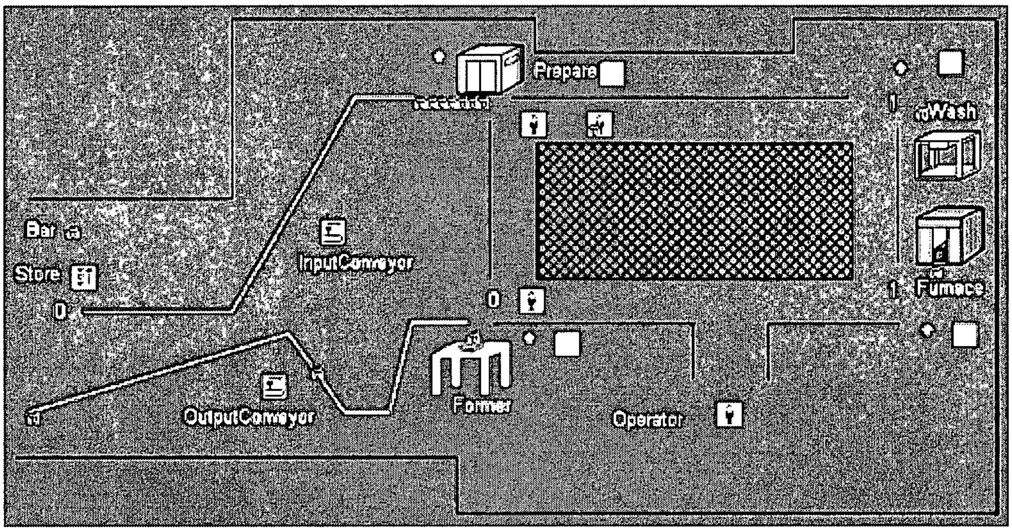


Figure 41: Representation of a model using Witness

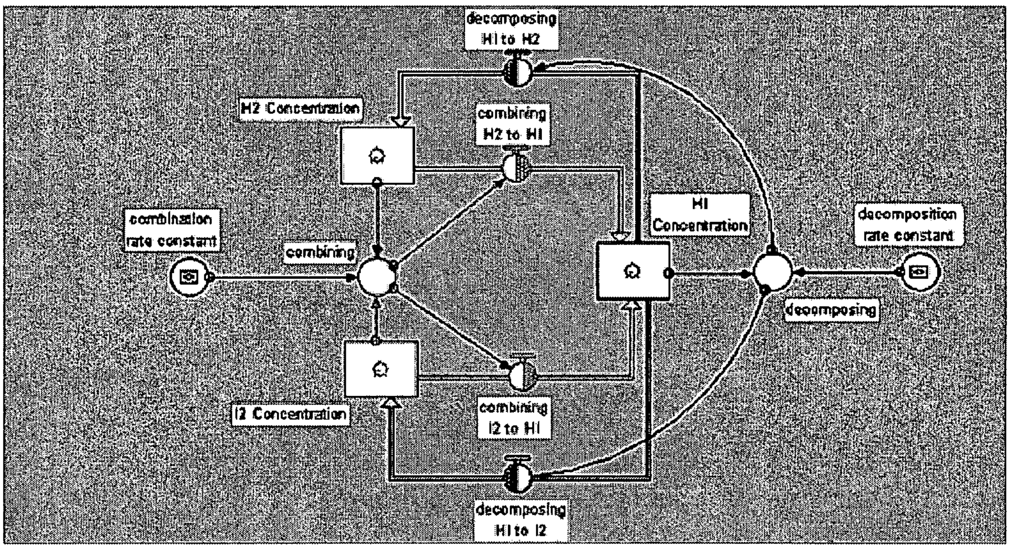


Figure 42: Representation of a model using Stella/iThink

This analysis has revealed that many similarities exist between tools that belong to the same modelling technique; however, differences in the working environment exist between tools from different techniques. For example, while SD based software attempts to offer as few elements as possible, DES based tools offer many pre-built elements to facilitate the modelling task. However, all tools analysed offer menus for easy access to the built features, button bars for allocating the most used commands, and the modelling elements.

Thus, the final design of the user interface is based on inputs from literature, user needs, and analysis of existent tools. As stated in Section 5.1.2.1, a MDI consists of a parent form and one or several child forms. However, the way in which these forms are interconnected relies exclusively on the programmer. As stated by Mullet and Sano (1995), *"the usability of a computer programme depends on both the design of the individual forms that compose the programme and on how they are connected"*. Thus, the interface design allows the user to flow through the different forms in an intuitive and simple manner. In addition, and following the recommendation of Pidd (1988), simple icons have been used to facilitate the learning curve.

The final design of the tool is based on the steps presented in Section 2.4.1, where the main stages of a simulation study were explained. In this process, tasks such as: (i) Model building, (ii) Data acquisition, (iii) Model execution and (iv) Results of analysis (through statistics and graphics) were identified. Thus, the design of the tool consists of a parent form, containing five main child forms and some accessory forms. The structure of the interface is shown in Figure 43 (full page screenshots are provided in Appendix E), while detailed information about each form is provided below

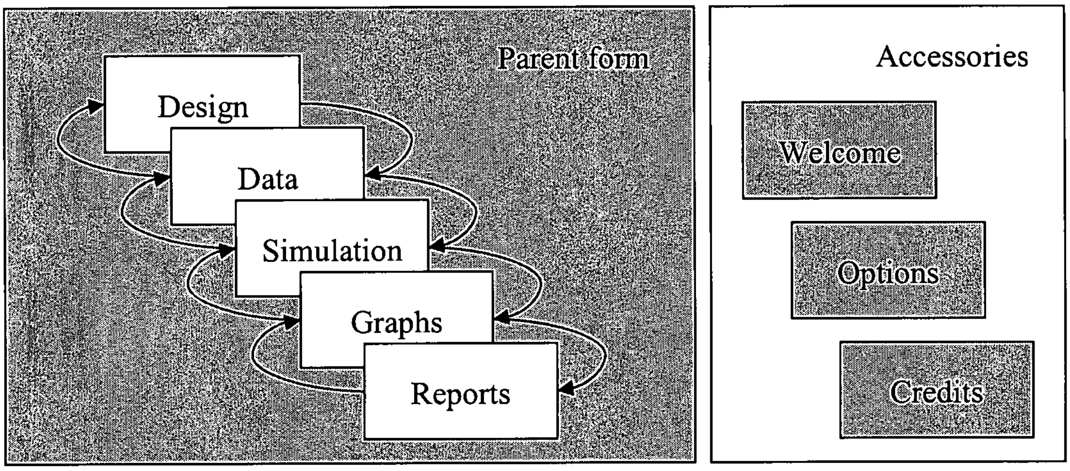
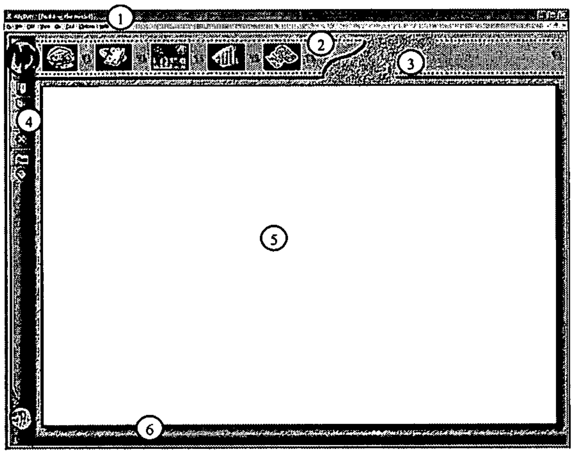


Figure 43: Interface structure

Parent form – The container: This form contains standard components to all the child forms, as well as personalised components depending on the child form displayed. A screenshot of the parent form is shown in Figure 44, while the main characteristics are explained below.

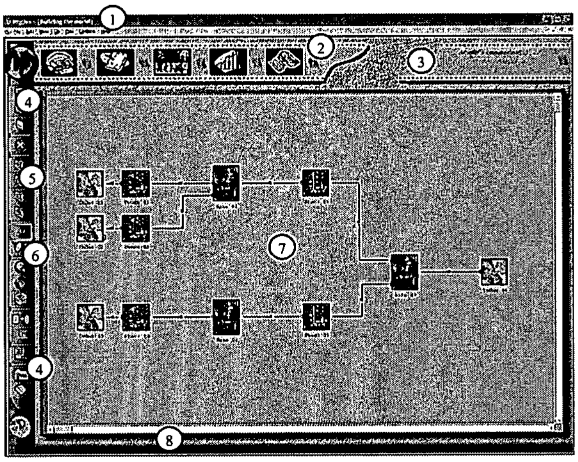


1) Menus
2) Forms navigator
3) Model statistics
4) Basic buttons
5) Personalisation area
6) Status bar

Figure 44: Screenshot of the container form

- Menus (1): Includes the functionality provided and shortcuts (when available) for the form that is being visualised. For example if the model building form is visualised, the available menus are:
  - File (New, Open, Save, Save as …, Print screen, Exit)
  - Edit (Undo, Cut, Copy, Paste, Delete)
  - View (Zoom +, Zoom -, Zoom 100, Zoom adjust, Process flow)
  - Go (Design, Detail, Simulation, Graph, Report)
  - Options (Model, Initialise actions, Display, Find)
  - Help (Contents, Contact the author, Web support, About MfgDyn)
- Forms navigator (2): Provides a link to the desired form (design, data, simulation, graphs and reports).
- Model statistics (3): Shows basic information dependent on the active form. For example in the model building form, it displays the number of different elements inserted into the model.
- Basic buttons (4): New, Open, Save, Exit, Configuration and Help are included as basic buttons.
- Personalisation area (5): Is the area used by the sub-forms to display the specific area of a simulation stage (design, data, simulation, graphs and reports).
- Status bar (6): Provides help and guidelines to aid the modelling task.

Child form 1 – Model building: As it name indicates, the model building form is used by the user to develop the computer models of a system. A screenshot of this form is shown in Figure 45, while the main characteristics are described below.

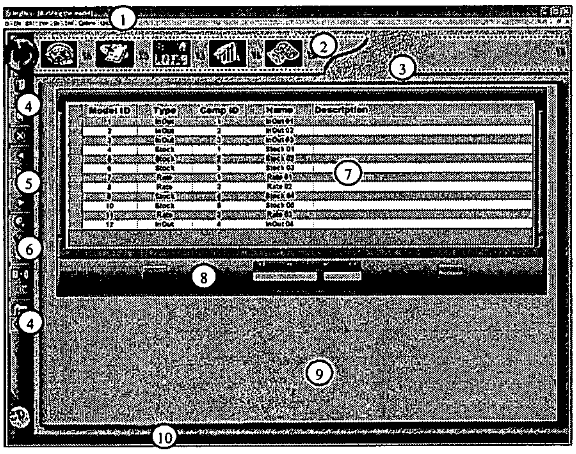


1) Menus
2) Forms navigator
3) Model statistics
4) Basic buttons
5) Visualization buttons
6) Modelling buttons
7) Modelling working area
8) Status bar

Figure 45: Screenshot of the model building form

- Modelling buttons (6): Includes a pointer (used to locate, move and delete the elements), a 'link' element (used to create the flows of materials between the elements), and the predefined elements to construct the model. These are: (i) Parts (called 'InOut', represent the material that enter/exit the system), (ii) Stocks, (iii) Rates (represent the machines), (iv) Converters and (v) Constants.

Child form 2 – Data introduction: In this form, data related (attributes) to the elements previously built is introduced. A screenshot of this form is shown in Figure 46, while the main characteristics are described below.



Model ID	Type	Comp ID	Name	Description
1	InOut	1	InOut 01	
2	InOut	2	InOut 02	
3	InOut	3	InOut 03	
4	Stock	1	Stock 01	
5	Stock	2	Stock 02	
6	Stock	3	Stock 03	
7	Rate	1	Rate 01	
8	Rate	2	Rate 02	
9	Stock	4	Stock 04	
10	Stock	5	Stock 05	
11	Rate	3	Rate 03	
12	InOut	4	InOut 04	

1) Menus
2) Forms navigator
3) Statistics
4) Basic buttons
5) Navigation buttons
6) Shortcuts
7) Summary of model's elements
8) Navigation and element's status
9) Properties area
10) Status bar

Figure 46: Screenshot of the data introduction form

- Navigation buttons (5), Shortcuts (6) are used to navigate through the different elements of the model.
- Navigation and element's status (8) is displayed if changes to a particular element have been produced, and allows the user to save/discard them.
- Summary of model's elements (7) displays all the elements introduced in the model and basic information, such as their name, type and a description.
- Properties area (9) is used to specify the attributes of the elements. This sub-form is different regarding the selected element (InOut, Stock or Rate) (see Figure 47). Functionality of these elements is presented in the next section.

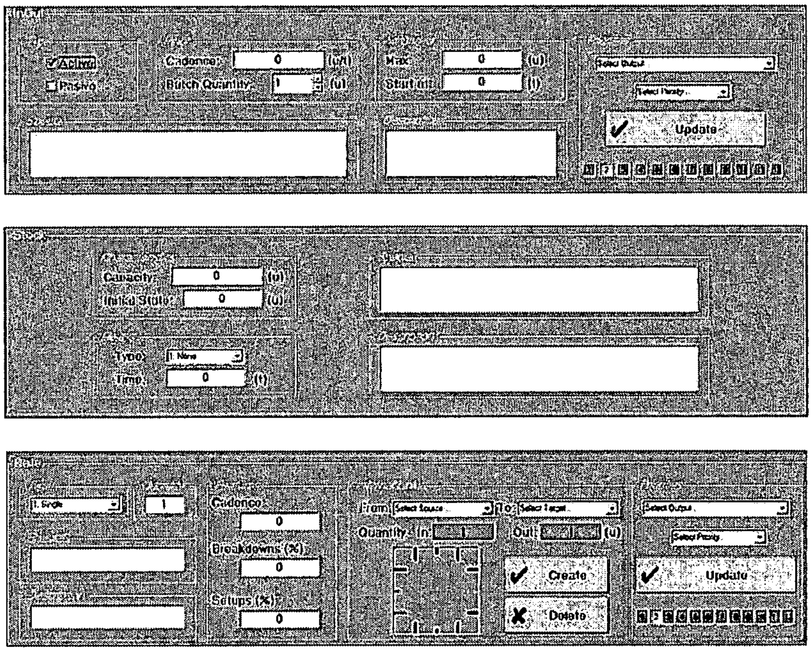
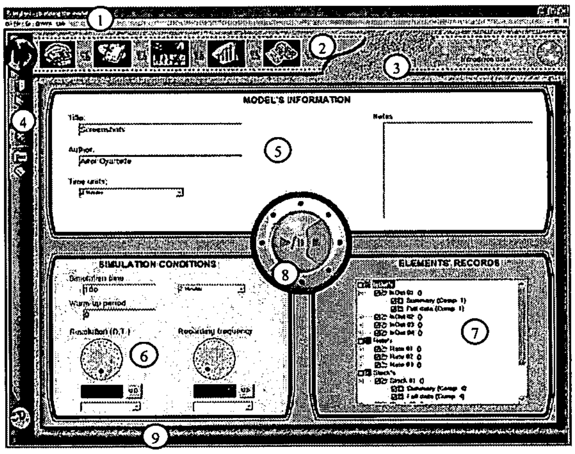


Figure 47: Screenshot of the a) 'InOut', b) 'Stock' and c) 'Rate' properties sub-forms

Child form 3 – Model execution: This form is used to introduce the simulation conditions and the elements that need to be monitored and run the model. A screenshot of this form is shown in Figure 48, while the main characteristics are described below.

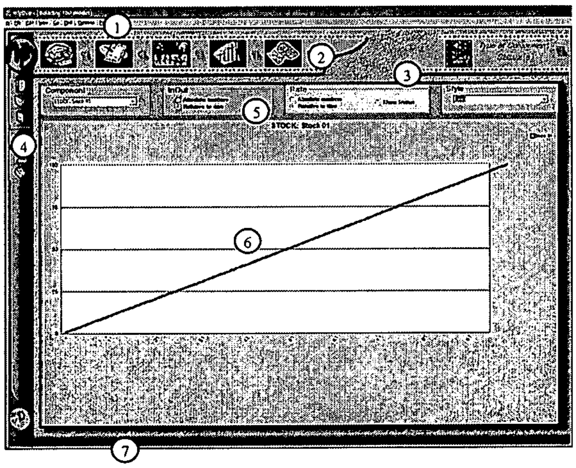


1) Menus
2) Forms navigator
3) Monitoring
4) Basic buttons
5) Model's information
6) Simulation conditions
7) Element's records
8) Start/Pause/Stop
9) Status bar

Figure 48: Screenshot of the model execution form

- Model's information (5) includes basic information about the system modelled as well as the time units used.
- Simulation conditions (6) define the simulation horizon as well as the warm-up period (which will be excluded from the statistics). Then, the resolution of the simulation must be set (DT), together with the recording interval (if lesser level of detail is required).
- Element's records (7) specifies the elements of the systems that need to be monitored; this data will be used by following forms (graphs and reports) to create the output screens.

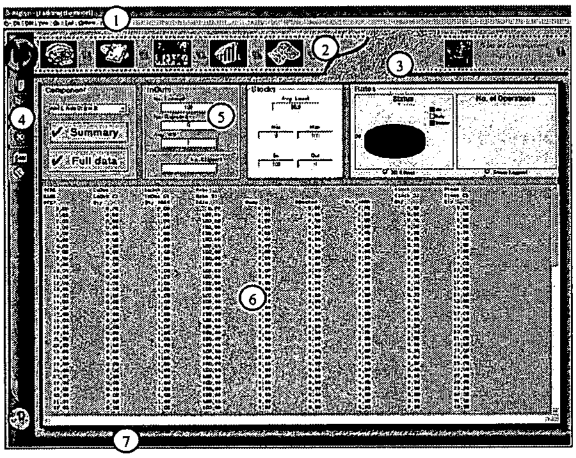
Child form 4 and 5 – Graphical and numerical results: The final two forms are used to visualise the resulting graphs and reports, respectively. Screenshots of these forms are shown in Figure 49 and Figure 50, while the main characteristics are described below.



1) Menus
2) Forms navigator
3) Statistics
4) Basic buttons
5) Graphics selector
6) Graphics
7) Status bar

Figure 49: Screenshot of the graphical results form

- Graphics selector (5), as it names indicates, selects the graph to visualise the preferred display method.



1) Menus
2) Forms navigator
3) Statistics
4) Basic buttons
5) Reports selector and summary
6) Reports
7) Status bar

Figure 50: Screenshot of the numerical results form

Reports selector and summary (5) displays the name of the element that is being visualised and a summary of the information related to it. If full data are required, it can be visualised in the reports area (6).

In addition to the main forms mentioned above, the tool includes three accessory forms that are used to: (i) Welcome the user, (ii) List the options available at the beginning and (iii) the credits of the software and contact details. A screenshot of these forms is shown in Figure 51.

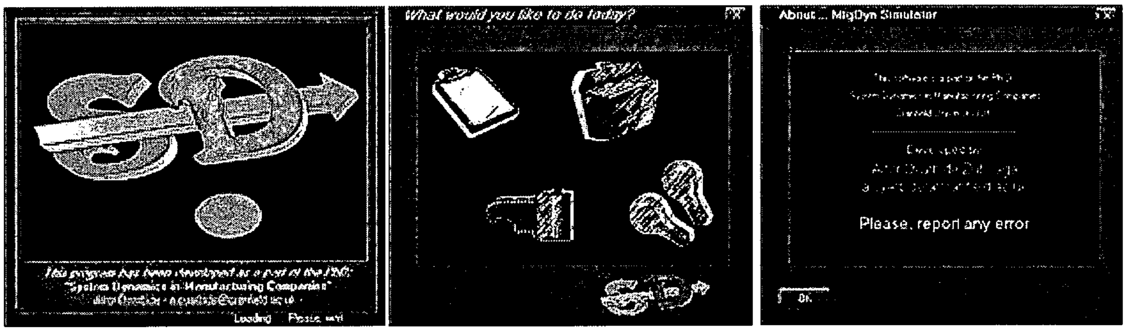


Figure 51: Screenshot of the a) Welcome, b) Options and c) Credits forms

5.3.2 Analytical capabilities of the tool

The previous section has defined the user interface and interactive capacity. This section complements the graphical design of the tool by adding analytical functionality. Thus, this section justifies the selected attributes for the elements integrated in the tool (parts, stocks and rates) and the algorithms implemented for obtaining the selected performance measures.

5.3.2.1 Element's attributes

Literature has identified parts (in the nature of flows), stocks and rates as the basic elements of the SD technique for constructing models. This was corroborated by the survey conducted in the first stage of the research, where all interviewed members found the inclusion of 'parts (flows)', 'buffers (stocks)' and 'machines (rates)' essential. Thus, Section 5.2 selected these elements for implementation within the tool. The properties associated with the selected elements, as well as the assumptions made to implement them in the tool, are described below:

Within the SD technique, 'parts' are used as in/out elements. These elements introduce flows of parts into the system or take them out. If the element is used to record the quantity of material that flows out of the system, no properties are required; however, if the element is used to introduce flows of material into the system, the following attributes can be adjusted:

- Rate (compulsory): Determines the number of parts that enter into the system per unit of time.
- Maximum quantity (optional): Indicates if the flow of material needs to be stopped after reaching a predefined level.
- Start at (optional): Specifies when the flow of materials starts.
- Priority (optional): A priority can be set if the flow of parts is linked with more than one element of the system. Within the tool, 'priority' has been implemented in a discrete manner to simplify the code. However, the SD technique allows the user to model priorities by simply increasing/decreasing the control functions. Thus, the essence of the SD technique is maintained.

Stock (or level) is used to store the parts. Because of the continuous nature of flows within the SD technique (see Section 2.2.2), stocks are only allowed to store one type of flow. Thus, it is not necessary to include complex control algorithms, since flows of materials within stocks are controlled based on 'First In First Out (FIFO)'. The main attribute of this element is 'capacity', which defines the maximum number of parts that can be stored in stock. All DES tools analysed in the previous section also included the possibility for defining an initial state for buffers, indicating the number of parts stored in the buffer at the beginning of the simulation. Due to the compatibility of this feature with the SD technique and the usefulness in practice (to simulate, for example, preloaded systems), this attribute has also been included.

As stated in Section 2.2.2.1, a quantitative SD model is *"a set of difference equations whose variables change their value through time"* (Pidd, 1988). Wang and Skeel (2003) analyse several numerical integration methods, such as 'rectangular', 'trapezoidal', 'Simpson', etc., to integrate the difference equation using a computer. In summary, the accuracy of integration is related to the complexity of the algorithms; thus, accurate algorithms, such as Runge-Kutta, require significantly more calculation time than simpler methods, such as Rectangular. Following the suggestions of Wang and Skeel (2003) and the purpose of the tool, the 'Rectangular method' has been selected for numerical integration, since it is a good compromise between accuracy and calculation requirements.

Rate (or machine) controls the material flows of the model by processing raw material and producing products at a predefined rate. Standard information to include in all types of machines is: (i) Cadence, (ii) Breakdowns (as a %) and (iii) Setups (also as a %). In addition, flows must be defined (and priorities can be set) when more than one product is manufactured in one machine.

Within the SD technique, control can be of various forms, such as proportional and differential. Forrester (1969) suggested a simplified control method based on first-order feedbacks (see Section 2.2.2); its inclusion in the tool was justified in Section 5.2.5.1. This method, also called 'proportional control' (Powell *et al., 2001)* is commonly used in SD models (Forrester, 1961 and Sterman 2000). It expresses the control signals in the form of equations, such as:

➢ $PC_t = a * (MCT_t - TCT_t)$ (Power *et al.,* 2001)

where:

— PC = Proportional adjustment

— MCT = Measured Cycle Time

— TCT = Target Cycle Time

— a = Constant proportionality (also called proportional gain)

— t = Time

The application of this equation in practice can create situations of 'over-saturation' or 'mis-utilisation'. For example, if a low value is selected for the parameter 'a', the value of PC can be so small that the machines never reach their maximum capacity; conversely, high values of 'a' can provide values of PC that force machines to work over their limits. Thus, to match the behaviour of the modelled machines with the real ones, a very high value of the 'a' parameter is selected (ideally infinite) within this tool and the value of PC has been limited to the maximum capacity of the machine.

In order to make machines more user friendly, and to more closely match industrial needs, machines can be tailored to match typical situations. Thus, and based on commercial DES tools such as Witness, the different machine types implemented are:

1. Single machine: The machine takes one 'part' and delivers one 'product'.
2. Assembly machine: The machine takes several 'parts' and delivers one 'product'.
3. Production machine: The machine takes one 'part' and delivers several 'products'.
4. General machine: The machine takes several 'parts' and delivers several 'products'.

5.3.2.2 Performance measures

Section 5.2.5.2 identified 'machine rate', 'machine utilisation', 'throughput or product throughput' and 'work in progress' as required performance measures to be included in the tool. These measures are calculated by applying the SD concepts to the elements associated with them. Thus, to obtain 'machine rate' and 'machine utilisation', the production rate of the machines needs to be analysed; whereas parts leaving the system and stocks need to be analysed to calculate 'throughput' and 'work in progress', respectively.

- Machine rate: In DES simulation, if a machine is configured to conduct a task, it can only start to process the materials if the quantity required is available beforehand. However, if a machine is modelled using the SD principles or if a machine is configured to conduct a task, it can start processing flows of materials even it they are lower than the maximum amount of material that the machine can process per unit of time (the machine will simply process the flows of material more slowly). Thus, calculation of machine rate is done by recording the instantaneous machine cadences and calculating the average for the period specified.

$$MR_{av} = (\sum_{T1}^{T2} MR_t) \cdot \frac{DT}{T2 - T1}$$

where:

- MR_{av} = Average Machine Rate
- MR_t = Machine Rate at the time 't'
- DT = Delta Time
- T2 – T1 = Time interval considered

- Machine utilisation: Machine utilisation is defined as the percentage of the capacity being used by a machine. Thus, this measure is obtained from the previous one. For a given interval, the machine utilisation is the average machine rate divided by the maximum machine rate.

$$MU_{T2-T1} = \frac{MR_{av(T2-T1)}}{MC} \cdot 100\%$$

where:

- MU_{T2-T1} = Machine utilisation for the period T2-T1
- $MR_{av(T2-T1)}$ = Average machine rate for the period T2-T1
- MC = Machine capacity

- Throughput: Represents what the system produces and is calculated by adding all the components that arrive to an 'output' part. The result can be given as an absolute value (for example, 1500.23 parts have been produced) or as an average (for example, 12.25 parts per hour are produced). Note that both values can be 'real (with decimals)' because SD treats products as flows.

- Work in progress: This value is calculated in the buffers. It records the instantaneous stock values each DT. Then, an average value for a period of time can be calculated using a similar procedure as the one to calculate the average machine rate.

In addition to the performance measures provided directly, other performance measures can be calculated indirectly. As stated in Section 5.2.5.2, all data gathered during the execution of a model is stored in an Excel file. Thus, by manipulating this data manually, different performance measures can be obtained (for example, an estimation of the 'lead time' or 'queue length' of a product, identification of bottlenecks, etc.).

5.4 Implementation of the tool, integration and tests

The implementation of the tool was done using Microsoft Visual Basic as the programming language (its justification is reasoned in Section 5.1.3.3). It was implemented using the specifications provided in Section 5.2 and the design proposed in Section 5.3. Similar to the design process, the implementation was divided into two main stages. The first stage consisted of the implementation of the graphical capabilities (structured as shown in Figure 43). Then, the functionality associated with each form was embedded in a modular form, as suggested by Petroutsos and Hough (1999). They state that the layout of the code can have a significant impact on the ability to read it later, and provide the following example:

> *"While it may be perfectly valid to pack five Visual Basic statements on a single line or to write a procedure with no indentation, the code will be nearly unreadable in the future, even by the developer."*

Although the tool developed in this research includes only a part of the functionality expected from commercial simulation tools, approximately 9,000 lines of code were needed. Thus, it appears apparent that a structure was needed to carry out this task successfully. The following guidelines (suggested by Petroutsos and Hough, 1999) have been followed during the development of the code, thereby making the software readable and allowing easier expansions in the future.

- Develop small modules that can work independently, and link them at the end. This reduces the error rate, testing time and size of the executable file. In addition, it increases productivity and reusability.
- Select the right type of variables. To improve the execution time, 'Variant' type of data have been avoided. Variants are generally Visual Basic's slowest data type. Execution time has also been benefited by using 'integer' variables instead of 'real' ones.
- Follow a convention for declaring the name of variables and constants.
- Place a block header comment at the top of each module and procedure.
- Indent all the code within a procedure and indent all loops.
- Add inline comments to describe data declarations and blocks of code

As an example of code procedures, a sample of the code implemented in the 'model design' form is shown in Appendix F, and an overview of the programming code responsible for executing the model is presented in Figure 52. As can be appreciated in Figure 52, when the 'Start simulation' button is pressed in the tool, the system checks if the modelled elements contain the attributes required to conduct the simulation. In the affirmative case, the simulation starts; otherwise, it is aborted. When the simulation starts, the system first checks the links between the elements to define the flows. Then, attributes related with the elements are loaded in memory and initial values are updated. Once flows are defined and all values are set, Delta Time (DT) is set to its next value, and the calculations are repeated. This operation continues until the predefined simulation horizon is achieved, to finally report the data.

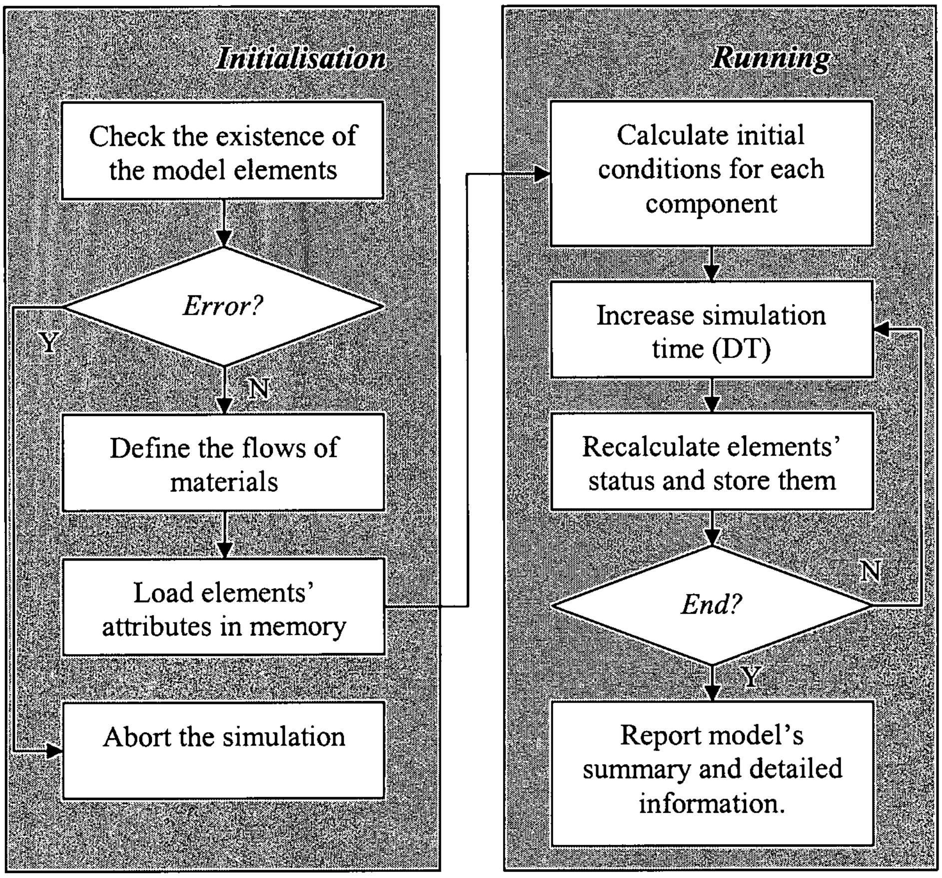


Figure 52: Flow chart for running procedure

After developing the tool, it was validated following the process described by Ellman (2000), as discussed in Section 3.5.3.2. Thus, the user interface was tested to validate its functionality. In addition, each individual procedure implemented in the tool was tested by introducing a predefined set of data (Zelkowitz *et al.,* 1979) and results were analysed. Then, an 'integration test' was conducted to test the interactions between the different modules and data. This test also included the modelling of a simple model; it consisted of ten parts that flow through a buffer, are processed by a machine, and then leave the system after passing through a second buffer (see Figure 53). The results obtained by the tool were compared with results calculated by hand; identical results where obtained.

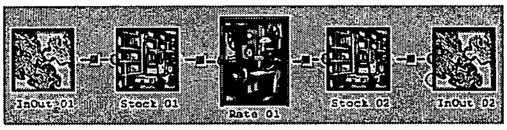


Figure 53: Validation of the tool (example model)

The final test suggested by Zelkowitz *et al.* (1979) is called 'system test'. In this research, a simplified 'system test' was conducted using the model shown in Figure 53, in order to evaluate the execution speed of the tool, and benchmark it against a popular SD commercial tool. This test was motivated by the nature of the developing tool used to build the tool, namely Visual Basic (VB). Section 5.1 mentioned that programs developed using VB are usually slower than programs developed using other programming languages, such as C++. Considering that the developed tool is going to be compared against a commercial DES tool (Witness) in the next stage of the research, this test was conducted to determine the extent to which the developed tool or the SD technique were responsible for the time required for execution. The results obtained from this test (both models were executed five times in the same computer) revealed that the developed tool was 5.88% slower than the commercial SD tool. This difference, although noticeable, can be considered acceptable for the purpose of this research.

5.5 Summary

This chapter has presented the development of a tool based on the SD principles. The requirements for this tool were gathered from the literature review carried out previously, data obtained from the interviews conducted in stage one of the research and the scope of this research. The second part of this chapter detailed aspects related to the design of the tool, including both visual and analytical capabilities. In this section, screenshots of the tool were provided, and the mathematical background needed to obtain the specified performance measures was overviewed. Finally, aspects related to the implementation of the tool and testing are presented. The tool was validated by testing its procedures individually, their integration within the tool, and by modelling a simple model. While accuracy of the results was validated successfully, the tool obtained the expected results but the execution speed was found to be slightly slower than a commercial SD based tool.

Although the tool developed is in the prototype stage, this chapter has demonstrated that the principles of SD can be implemented in an Object-Oriented (O-O) tool with iconic representation. It will be used in the next stage of the research to conduct the case studies and to compare different assessment criteria.

Chapter 6

DEVELOPMENT, EXECUTION AND RESULTS OF A TEST BED FOR ANALYSING THE SUITABILITY OF SYSTEM DYNAMICS

Chapter 5 described the second stage of the research; designing and developing a computer tool able to simulate manufacturing processes applying the concepts of System Dynamics and using a friendly user interface. This chapter addresses the third research objective by designing and executing a number of case studies in order to assess the true capabilities of SD in a real environment. The structure of this chapter is as follow:

1. Case study method
2. Case study design
3. Execution and results
4. Analysis of results and key findings

6.1 Methodology for case study research

The multiple case study approach appears most likely to replicate the actual modelling challenges faced by practitioners. The justification for including the approach in this research is reasoned in Section 3.5.4. This section describes in further detail common issues related with case study research that will be used in the next section.

Case is defined by Gillham (2000b) as: *"a unit of human activity embedded in the real world; which can only be studied or understood in context; which exists in the here and now; and that merges in with its context so that precise boundaries are difficult to draw".* Thus, a case study is one that investigates the above to answer specific research question and which seeks a range of different kinds of evidence (Gillham, 2000b).

The case study method is detailed by Yin (1994). According to Eisenhardt (1989), a case study can be analysed following a: (i) Within-case analysis and (ii) Cross-case analysis. This phase of the research starts by adopting 'within-case analysis' as recommended by Eisenhardt (1989) and accepted by Voss *et al.* (2002). Reasons for this adoption are provided by Eisenhardt (1989) and Voss *et al.* (2002), who state that it is necessary to:

> *"... become intimately familiar with each case as a stand alone entity, and to allow the unique patterns of each case to emerge before you seek to generalise across cases"* (Eisenhardt, 1989).

> *"Cross-case analysis should seek to increase the internal validity of the findings (as emergent from within-case analysis)"* (Voss *et al.*, 2002).

The case study method (Yin, 1994) prescribes that the process should begin with the development of a theory; in the case of this research, the basis is the literature review presented in Chapter 2, the simulation requirements gathered in Chapter 4, and the design and development of the tool explained in Chapter 5. Then, issues such as: (i) Number of cases used, (ii) Selection of cases and (iii) Design of the collection protocol must be addressed prior to the execution of the experimentation programme (Yin, 1994). Baines (1994) and Adesola (2003) indicate key issues involved with the design of a case study, which include aspects such as:

- Experiment design
- Experiment control
- Choosing industrial test-beds
- Analysis methods
- Application of experiments

These issues, amongst other considered, are described in further detail in the following sections.

6.2 Case study design

Section 6.2.1 describes the performance assessment criteria. The selection of industrial cases (and its number) is explained in Section 6.2.2. Section 6.2.3 describes the role of the researcher and model builders. The data collection procedure is described in Section 6.2.4, whereas the model building procedure is explained in Section 6.2.5. Finally, Section 6.2.6 describes the analysis procedure.

6.2.1 Performance assessment criteria

Case studies aim to assess the suitability of each technique and the tools involved in different real situations. In this context, suitability is defined by considering both quantitative and qualitative practitioners' expectations. Thus, when both methodologies offer similar results when dealing with a particular set of requirements, different qualitative features such as model building time will be balanced in order to identify the strengths and weaknesses for each technique.

Smith (1990) provides useful advice on the conduction of case studies, suggesting that a successful approach to complex case studies is to break them up into smaller sub-cases, and to complete the work in parts. In addition, the selection of adequate test-beds is important. In line with the work of Baines (1994), four key assessment criteria were identified, namely:

1. Accuracy: If real data are available, the results obtained by the developed tool (MfgDyn) and the commercial DES based tool (Witness) are compared against real data. Otherwise, results obtained are compared one against the other.
2. Modelling time: Quantitative and qualitative data about the time required to build the models will be gathered and compared.
3. Execution time: Models created in both MfgDyn and Witness will be executed in identical computers, and time will be recorded and compared.
4. Required skills: This is a qualitative measure that measures the help required by the 'model builders' to conduct the modelling task, together with their opinions.

6.2.2 Selection of industrial cases

As stated in Section 3.5.4.1, Eisenhardt (1989) suggests a guide as to the number of cases that a typical study should conduct; he states *"A number between four and ten usually works well"*. However, Yin (1994) says that, since sampling logic is not applicable to case study research methods, sample size is irrelevant; instead, researchers should consider the number of replications they would like to be included. The number of replications varies in practice, and depends upon the certainty the researcher wants to have about the multiple-case results (Yin, 1994). In addition, the number of replications is also concerned with the nature of the research. Yin (1994) also provides an example, which states:

> *"For example, you may want to settle for two or three literal replications when the rival theories are grossly different and the issue at hand does not demand an excessive degree of certainty".* (Yin, 1994)

Based on the variety of parameters that are evaluated and the guidance provided by Yin (1994), three case studies were considered to be sufficient for the assessment of the SD methodology and were consequently undertaken during this research study. The main basis for deciding to conduct no more than three case studies was: (i) The results of the three case studies demonstrated an acceptable level of coherence with regard to the literature and information gathered in the first stage of the research and (ii) The scope of the tool prevents very complex systems, or systems with elements that vary from the implemented ones, from being modelled.

Baines (1994) provides some guidelines on choosing industrial test-beds. He argues that common types and sectors of manufacture are more appropriate for case studies, because the results obtained are relevant to more practitioners; this suggestion is followed in this research. As stated in Section 4.3, this research is funded by the Basque Government. Thus, the intention has been to use Basque companies for case study from the pool of firms participating in the first stage of this research (the interview survey). However, although two of the Basque companies that participate in the first stage were willing to participate further, a search for new companies had to be conducted. A typical letter requesting collaboration with potential case study firms can be found in Appendix G.

The criteria for case company selection stipulated the search for organisations that were SMEs. The reasons for this included the typical characteristics of their processes (usually smaller and less complex), their availability and relevance. In total, eight companies that showed interest in the project were visited; five of the companies were selected to conduct the case studies. After a first screening process, two companies were discarded due to the similarity of their processes with previously selected companies or unavailability of data. In order to keep the confidentiality of these companies and to avoid misunderstandings with the companies visited in stage 1 of the research, they were named Company X, Y, Z (see Table 33).

Company	Sector	Location
Company X	Home appliances	Arrasate, Spain
Company Y	Automotive supplies	Eskoriatza, Spain
Company Z	Automotive supplies	Arrasate, Spain

Table 33: Location and sector of selected companies

6.2.3 Role of the researcher and model builders

The researcher is actively involved through the experimentation process. In the first stage of the case study, the researcher is responsible for contacting the selected companies, analysing their processes and selecting an appropriate section to model and analyse and finally, collect the data.

The first requirement for any participant in case study research is to identify its purpose, so that it will not bias other members involved in the case study, or the case study itself (Gillham, 2000b). This is especially important in the second stage (model building) where the researcher is involved in the creation of the models in collaboration with other researchers. The purpose of conducting this stage of the research by the researcher and other members is to acquire more realistic data and opinions about the time spent in the different stages of model building. During this research, each of the three models to be analysed were modelled by the researcher in the developed tool, as well as in Witness (a relevant DES based commercial tool – see Section 2.4.3 and Section 4.3). In addition, to minimise the effect of 'learning curves', groups of two members were defined in order to conduct the model building and execution. While one of these groups first modelled the system using the tool developed in this research, and then using Witness, the other group followed the opposite approach.

The final stage of the case study involves analysing the results obtained by the case study, consistency with the literature, and possible implications within industry. This stage is conducted by the researcher.

6.2.4 Data collection procedure

An initial visit identified the company areas to focus the case study on. An important aspect of case research is to obtain a good understanding of the subject and the context under study; in this case, the processes being analysed and the data involved with them.

As stated by Gillham (2000b), data collection is a technique *"to be used sparingly: it takes time in planning, is very time-consuming, and yields limited information"*. Thus, prior to the data collection (and due to the inadequacy of disturbing companies unnecessarily), the researcher analysed the overall model shape gathered in a previous meeting, and enumerated the gaps. Then, in a second meeting with the company, detailed data was gathered with the help of an employee. The primary method used for data collection was face-to-face interviews with the contact member of the company, as well as other members when required. In some cases, data not available was forwarded to the researcher by post. Then, manipulation of data was conducted by the researcher in order to compile the obtained data into a suitable format that can be implemented in the tools.

Although data accuracy is desired, it was not crucial for the purpose of this research, because the same data (or manipulation of it) is used in both the SD and DES based tools. Thus, where data was collected by visual methods (for example, the cycle time of a machine), the selected sampling size was small.

6.2.5 Model building and experimentation procedure

Model building is concerned with transposing the conceptual model into a valid working model, using the modelling medium under consideration. Baines (1994) highlights the need for consistent model builder expertise to form comparable tests of modelling techniques. If, for example, the model builder has greater familiarity with one particular modelling technique or tool, it is likely that the progress made on model building will differ between the two tools used during the case studies. Furthermore, if the modeller approaches model building with different procedures in each case, it is likely that the application time will again be influenced.

Researchers involved in the case studies were students of Mondragon University. They assisted in a 24 hours simulation course in which models were built using Witness, and the researcher also provided them detailed information about the usability of the tool created during this research. In addition, all the researchers involved in this stage of the research were asked to follow the procedure for model building and experimentation, explained in detail in Section 2.4.1, where the main tasks identified were:

- Model building and coding
- Model verification and validation
- Model experimentation

6.2.6 Analysis procedure

Data obtained from the execution of the case studies were collected: (i) Quantitatively and (ii) Qualitatively. The analysis procedure began by conducting a case analysis. In this stage, accuracy, modelling time, execution time and required skills were analysed (see Section 6.2.1), by using the following inputs:

- Results obtained from the DES and SD models, as well as real data, were compared to analyse the accuracy of the tools.

- Each of the two teams involved in the modelling of a case study recorded the time spent to build the model. These measures, in addition to the time spent by the researcher, were compared.
- Model execution time was also measured, and the differences between the two tools were compared.
- Qualitative information was gathered during the model construction stage, and help needed by the students when conducting this task was also recorded.

Second stage of the analysis involved a 'cross-case analysis'. Thus, the findings obtained in each case analysis were collected in this stage to enable case similarities and dissimilarities to be drawn out.

6.3 Execution of cases and results

The three case studies (case X, Y and Z) are reported and considered here individually. Each analysis begins with a description of the company, followed by a brief description of the analysed process. Then, the proposed model (as well as the data required to model it) for the selected sub-section of the previously mentioned process is presented. This is followed by the proposed computer models created in Witness and MfgDyn (the tool developed in the previous stage). Finally, results regarding model building time, execution time, results from experiments and comments from the 'model builders' are presented.

6.3.1 Case X

Company X is a producer of home appliances, such as ovens, etc. The company was formed in 1956 in Arrasate (Spain), and now it is a part of one of the biggest industrial groups in the Basque Country. In 1999, the company had a turnover of 22,300 MPts (£95M approximately) and currently employs 148 people at its one site. Main products manufactured by this company include: (i) Ovens (representing approximately 50% of the total turnover), (ii) Washing machines, (iii) Dishwashers and (iv) Fridges.

This case study analyses the process selected to manufacture ovens, and more precisely, the stages involved to manufacture its decorative cover. This small section of the manufacturing system was selected because it has been designed recently and it is fully automated, using dedicated machines to manufacture the covers. Thus, the process is a linear flow (similar to the ones found in transfer machines). An overview of the main stages of the process can be seen in Figure 54 and is described below:

1. The manufacturing process begins with a stamping machine, where a piece of metal sheet is introduced, and a number of parts are manufactured.
2. Second operation consists of welding the parts produced in stage 1.
3. Then, an operator carries out a visual inspection, rejecting the components that do not meet the requirements. Due to the welding operations carried out in the components, rejected material cannot be re-introduced in the process.
4. Valid parts are then enamelled, in which both protective and decorative coatings are applied.
5. Stage 4 is followed by a second visual inspection, rejecting all parts that do not meet the requirements.
6. Final stage consists of drying the previously applied enamel. Then, the manufactured component is stored.

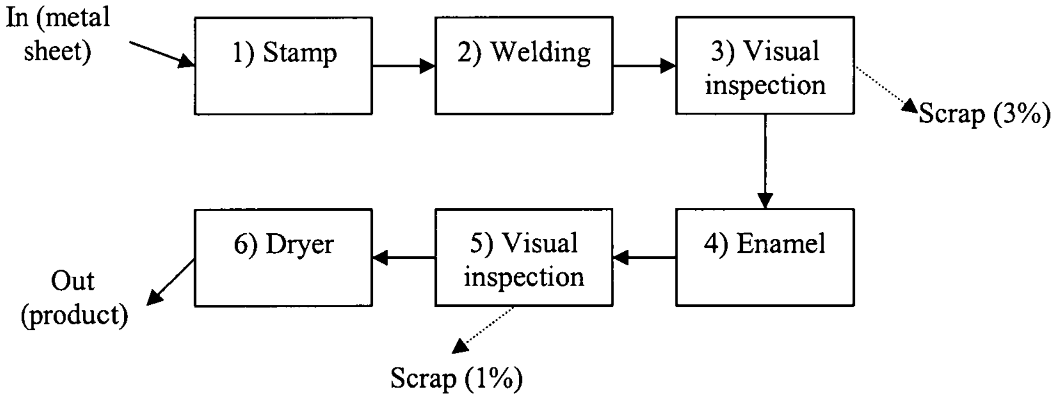


Figure 54: Flow diagram of the selected process (Case X)

Due to the recent implementation of the above flow, precise data about breakdowns was not available. However, average values were gathered by examinating maintenance reports. A summary of the data gathered for the construction of the model is shown in Table 34. In addition, the raw material (metal sheet) is considered infinite, because it is always available in practice.

Op.	Description	Cycle time (min)	Capacity (parts/h)	Stoppage (%)			Net capacity (parts/h)
				Setup	*Breakdown*	*Others*	
1	Stamping	0.353	170	8	3	9	136
2	Welding	0.349	172	1	9	10	137.6
3	Visual insp. 1	0.201	300	0	0	7	279
4	Enamel	0.316	190	1	3	11	161.5
5	Visual insp. 2	0.300	200	0	0	7	186
6	Drying	0.343	175	1	4	11	147

Stock	Initial value	Maximum value
Op 1-2	0	20
Op 2-3	0	80
Op 3-4	0	50
Op 4-5	0	30
Op 5-6	0	30

Table 34: Data collected from the system. A) Machines, B) Stocks

The next stage of the case study involved the development of computer models. In this case study, the same assumptions were made for both the tools (MfgDyn and Witness), because detailed data was not available. The sequence followed to develop the model was as follows:

- Researcher: Developed the model first in MfgDyn and then in Witness.
- Team A: Developed the model first in MfgDyn and then in Witness.
- Team B: Developed the model first in Witness and then in MfgDyn.

Screenshots of the model developed are shown in Figure 55 and the time spent in this task is provided in Table 35. In addition, comments provided by the two teams are given below.

Stage	Time (in minutes): a) MfgDyn, b) Witness		
	Researcher	*Team A*	*Team B*
Building the model	a) 2 min; b) 3 min	a) 4 min b) 3 min	a) 3 min; b) 5min
Detailing the model	a) 5 min; b) 6 min	a) 10 min; b) 10 min	a) 5 min; b) 8min
Verifying the model	a) 2 min; b) 2 min	a) 10 min; b) 1 min	a) 4 min; b) 2min

Table 35: Model building time (Case X)

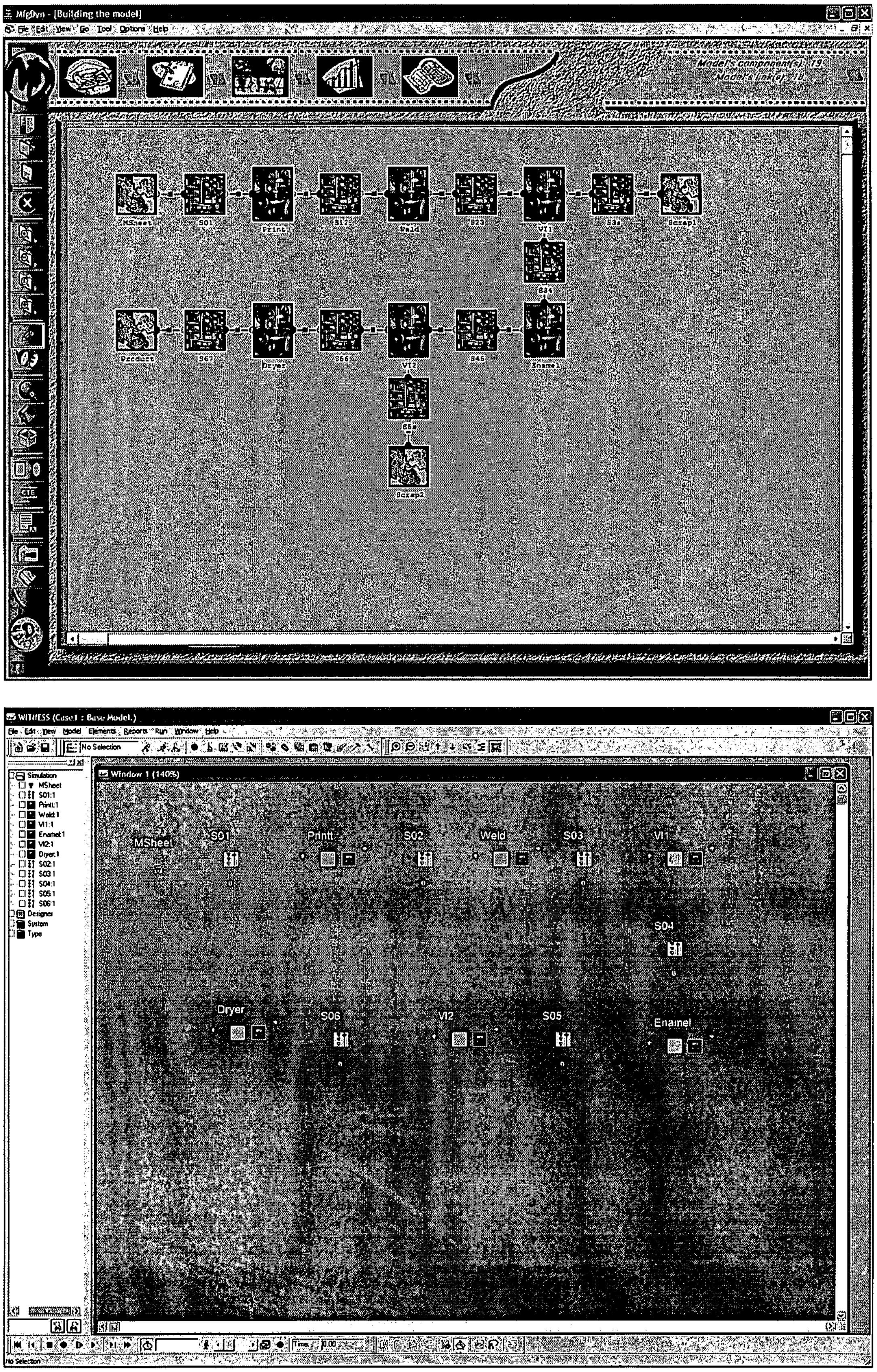


Figure 55: Screenshots of the computer model developed (Case X). A) MfgDyn, B) Witness

Main comments compiled from the model building stage are:

- Team A, B: *"MfgDyn's user-interface looks tidy, which avoids distractions."*
- Team B: *"MfgDyn definitely offers an advantage when creating flows. With only two clicks, two elements of the system get connected."*
- Team A: *"Adding details to the elements is a very simple task. All elements are in one screen."*
- Team A: *"Verification is faster in MfgDyn, because the attributes of all elements are in one screen. However, this stage could be improved if a summary of all data could be provided as a report."*

Final stage of the case study consisted of designing a set of experiments and comparing the efficiency and accuracy obtained from the two tools. The throughput (see Table 36 and Figure 56) and machine utilisation (see Table 37 and Figure 57) performance measures were selected; work in progress was not considered necessary because an analysis of the process revealed that the bottleneck is located in the first machine. Thus, if no breakdown data are considered, stock between the machines will always be minimal.

In addition, both tools were run over a long period of time in similar conditions (disabling the graphical capabilities of Witness). This experiment was designed to be able to compare the execution time of both tools (see Figure 58). A summary of the results obtained in this stage of the case study is provided below.

Simulation time	Results (number of products)				Difference (%)	
	Witness	MfgDyn			Max	Min
		DT=15s	DT=30s	DT=1min		
1min	0	0	0	0	0	0
2min	0	1.7	0	0	∞	0
5min	7	8.5	6.8	2.3	67.14	2.85
1h	132	133.6	131.4	126.6	4.09	0.45
2h	268	269.4	267.4	262.2	2.16	0.22
4h	540	541.0	539.3	533.4	1.22	0.13

Table 36: Throughput analysis (Case X)

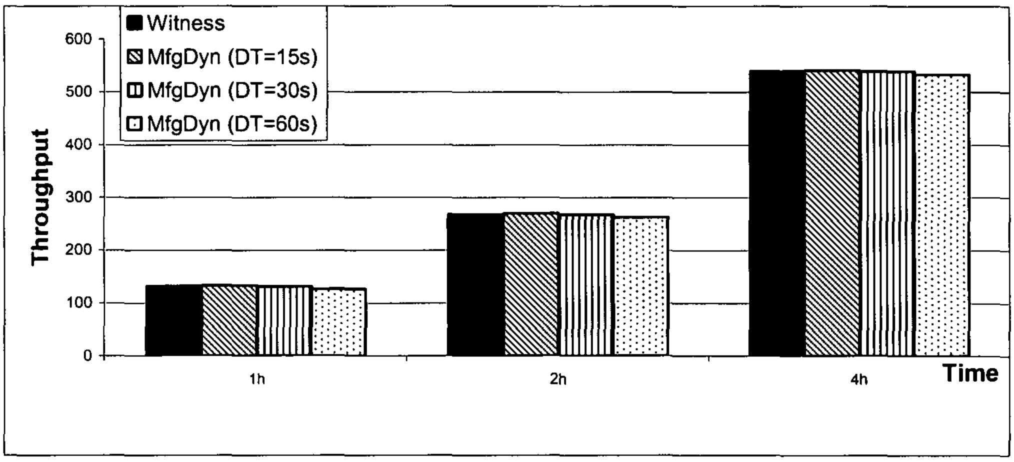


Figure 56: Throughput analysis (Case X)

Simulation time	Machine utilisation (%)			
	Witness	MfgDyn (DT=15s)	MfgDyn (DT=30s)	MfgDyn (DT=60s)
5min	a) 100; b) 41.78	a) 95.00; b) 46.30	a) 90.00; b) 34.12	a) 80.00; b) 19.50
1h	a) 100; b) 48.04	a) 99.58; b) 48.54	a) 99.17; b) 48.34	a) 98.33; b) 47.93
4h	a) 100; b) 48.57	a) 99.90; b) 48.69	a) 99.79; b) 48.64	a) 99.58; b) 48.54

Table 37: Machine utilisation analysis: a) Stamping machine (bottleneck) and b) Visual Inspection 1 (fastest operation) (Case X)

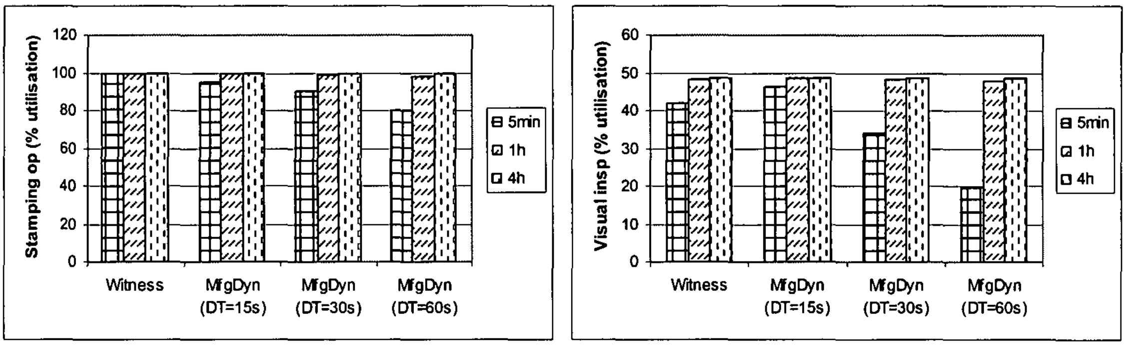


Figure 57: Machine utilisation analysis: a) Stamping machine (bottleneck) and b) Visual Inspection 1 (fastest operation) (Case X)

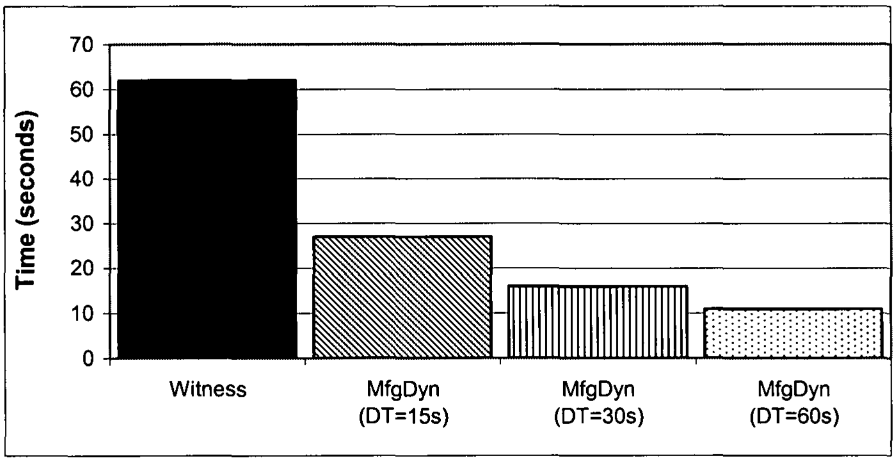


Figure 58: Execution time – 1 week horizon (80 hours) (Case X)

6.3.2 Case Y

Company Y is a producer of automotive supplies, specialised components made from aluminium and 'Zamak'. The company was formed in 1964 in Eskoriatza (Spain); in 2001, the company had a turnover of 500 MPts (£2.12M approximately) and currently employs 36 people at its one site.

Due to the small size of the company, it is focused on the manufacture of multiple products, but in small batches. This case study analyses the process selected to manufacture two aluminium components that shared the same line. Considering that the developed tool does not include a standard element to model batches (see Section 5.2), a simplification has been made, and the system is modelled only until the first series is finished. However, the analysis of this system is useful because it allows comparing the behaviour of the developed tool when priorities are present. Thus, the process consists of a linear flow (similar to the one described in Case X); however, two different components are manufactured, with the priority of one of them being higher than the other. An overview of the main stages of the process can be seen in Figure 59 and is described below:

1. The manufacturing process begins with a melting machine. At the beginning of the day, both 'Ref A' (2,048 units) and 'Ref B' (2,048 units) are stored in the buffer prior to the machine. Then, the predefined lot size Ref A is manufactured, followed by Ref B. The operation consists of melting the raw material, and casting eight components.

2. The second operation is manual, and consists of separating the previously cast parts, into separate pieces.

3. Final stage consists of cleaning the components. This is done in a 'vibration machine', where the previously manufactured parts are inserted in conjunction with the soap.

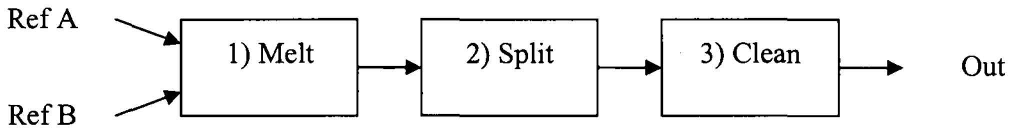


Figure 59: Flow diagram of the selected process (Case Y)

A summary of the data gathered for the construction of the model is shown in Table 38.

Op.	Description	Cycle time (min)	Batch quantity	Unitary CT (s)	Stoppage	
					Setup	*Breakdown*
1	Melt	0.675	8	5.06	60 min (A to B)	TBB: Normal (1970.4, 2124) MT: Normal (67.8, 96)
2	Split	0.360	8	2.7	0	0
6	Clean	30	256	7.03	15 min (A to B)	0
Note: Breakdown data for melting machine was provided by the company in the form of two normal distributions: Time Between Breakdowns (MTBF) and Maintenance Time (MTTR). The two attributes associated with each distribution are 'mean' and 'standard deviation', respectively.						

Stock	Initial value	Maximum value
Op 0-1a, b	0, 0	5000, 5000
Op 1-2	0	5000
Op 2-3	0	5000

Table 38: Data collected from the system. A) Machines, B) Stocks

The next stage of the case study involved the development of computer models. To simulate the breakdown distributions within MfgDyn a simplification was made and the percentage of time lost because of breakdowns was calculated, by dividing the MT by TBB. Thus, a stoppage of 3.2% was assumed. In addition, the setup time has to be simplified to be implemented within the MfgDyn tool. Thus, it has been considered that one setup change is done per shift (8 hours), and therefore stoppages of 12.5% (melting operation) and 3.12% (cleaning operation) have been assumed. The sequence followed to develop the model was as follow:

- Researcher: Developed the model first in Witness and then in MfgDyn.
- Team C: Developed the model first in MfgDyn and then in Witness.
- Team D: Developed the model first in Witness and then in MfgDyn.

Screenshots of the model developed are shown in Figure 60 and the time spent in this task is provided in Table 39.

Stage	**Time (in minutes): a) MfgDyn, b) Witness**		
	Researcher	***Team C***	***Team D***
Building the model	a) 2 min; b) 2 min	a) 5 min b) 4 min	a) 4 min; b) 5 min
Detailing the model	a) 4 min; b) 4 min	a) 7 min; b) 10 min	a) 6 min; b) 5 min
Verifying the model	a) 2 min; b) 2 min	a) 4 min; b) 3 min	a) 4 min; b) 2 min

Table 39: Model building time (Case Y)

The final stage of the case study consisted of designing a set of experiments and comparing the efficiency and accuracy obtained from the two tools. An analysis of the throughput curve was selected, to determine the extent to which the simplifications affect the behaviour of the model and to analyse whether the priorities are treated correctly within the MfgDyn tool (see Table 40 and Figure 61). In addition, the time required for manufacturing the 4,096 components (2,048 RefA and 2,048 RefB) was measured in both tools to present the differences in terms of accuracy. As in the previous case study, the model execution time was measured to compare the processing speed of the two tools (MfgDyn and Witness). A summary of the results obtained in this stage of the case study is provided below.

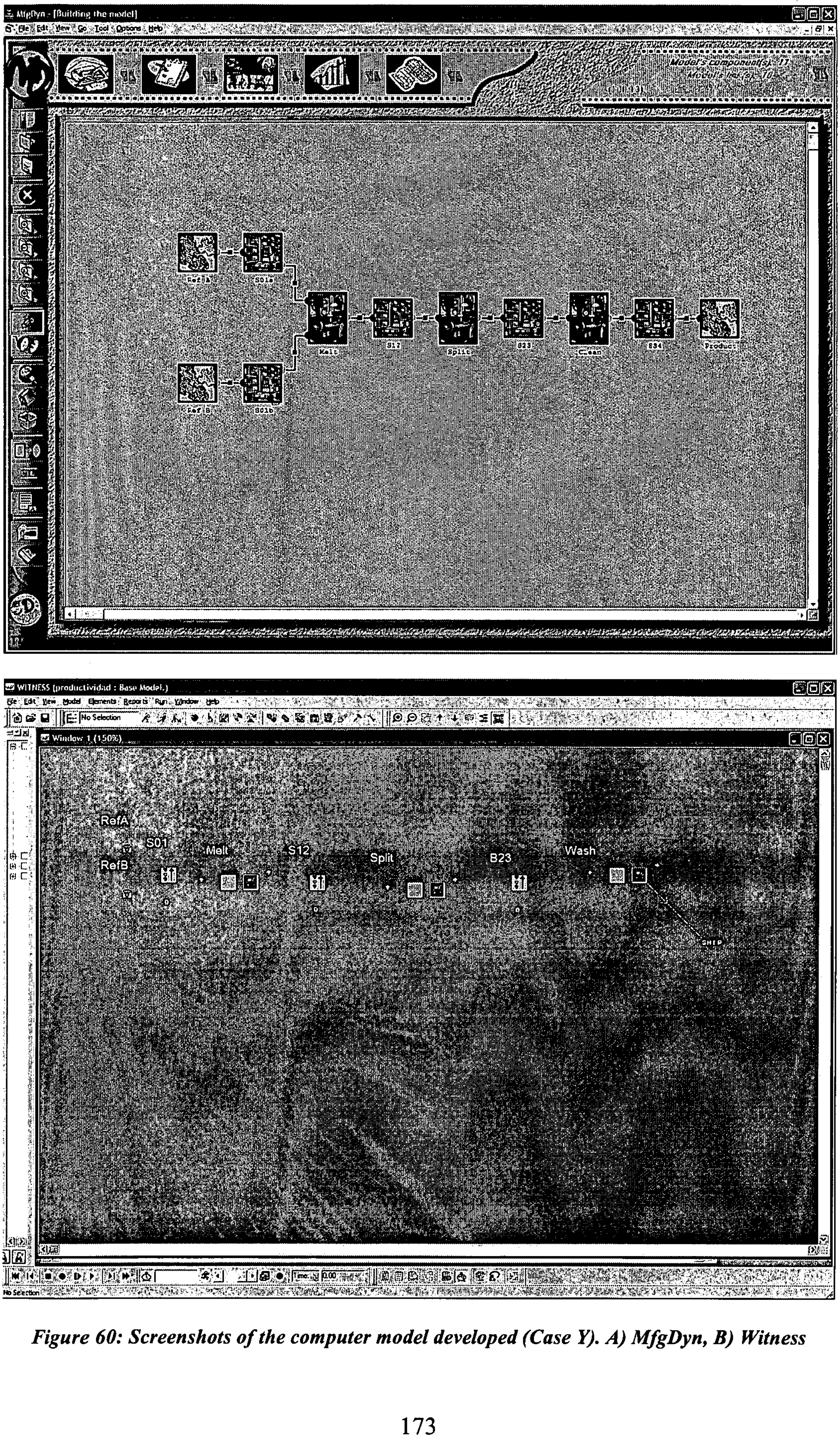


Figure 60: Screenshots of the computer model developed (Case Y). A) MfgDyn, B) Witness

Description	Results (number of products)			Difference (%)	
	Witness	MfgDyn		Max	Min
		DT=15s	DT=2 min		
RefA starts at	0	0	0	0	0
RefA ends at	262 min	248 min	252 min	5.25	3.8
RefB starts at	233 min	205 min	206 min	12.02	11.59
RefB ends at	517 min	496 min	500 min	4.06	3.3

Table 40: Throughput analysis (Case Y)

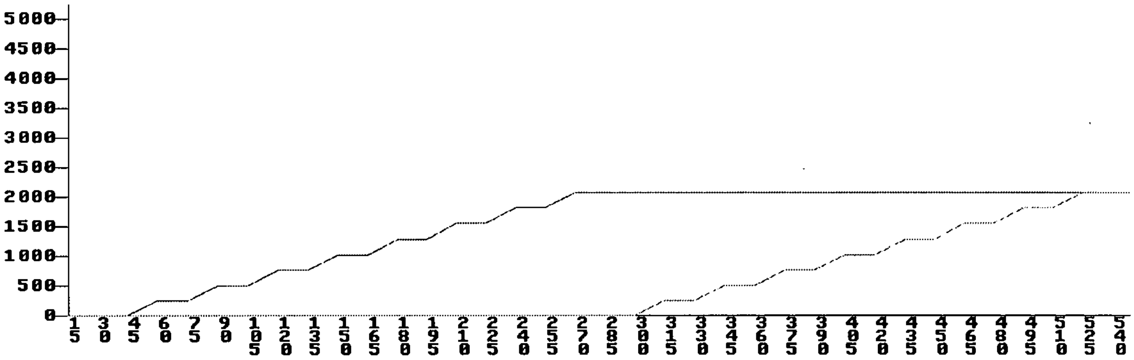


Figure 61: Throughput behaviour (Witness) (Case Y)

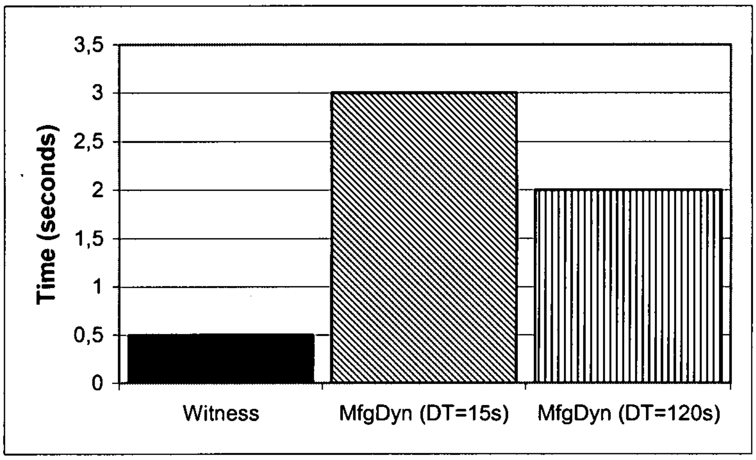


Figure 62: Execution time – 500 minutes horizon (approximately) (Case Y)

6.3.3 Case Z

Company Z is a producer of small metallic components, such as keys, rings, etc. The company is located in Oñate (Spain) and it is one of the oldest members of the Mondragon Corporation Cooperative (MCC) industrial group. In the year 2001, the company had a turnover of 311M€ (£217.7M) and currently employs 466 people in four different locations. The company currently produces a wide range of small metallic products, with expertise in stamping processes.

The case study conducted with this company is special. Although real data was acquired from the company, simulation was not used to obtain measures like throughput, work in progress, etc; but it was focused on discovering the advantages (or disadvantages) of the developed tool (and therefore, continuous simulation methods based on the SD methodology) as compared to DES based tools. This company was already using DES based tools to simulate their manufacturing systems. However, due to the very low cycle times of their operations (<5 seconds), added to unitary transfer lots (often) and the high frequency of very short breakdowns (for example, when a key is not ejected properly, an operator removes it manually), made execution of simulation a very time consuming task. Thus, an analysis of model execution time was conducted, by simulating first a single machine of the process, and then a transfer machine consisting of three operations. In order to analyse only the effect of model size in the execution speed, no setup times or breakdowns were considered. An overview of the two sub-processes analysed in this case study can be seen in Figure 63.

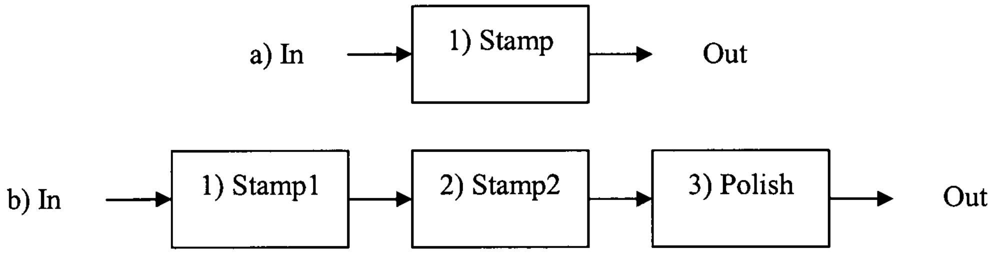


Figure 63: Flow diagram of the selected sub-processes (Case Z)

This model was coded and executed only by the researcher. This decision was based on the simplicity of the models and the objective of the case study. Table 41 and Figure 64 show the time required by both tools (MfgDyn and Witness) to model one month (720 hours) horizon.

Scenario	Execution time	
	Model A. Production rate = 12 parts/min	*Model B. Production rate = 14 parts/min*
Arena (1)	40s (2)	416s (3)
Witness	34s	64s
MfgDyn (DT=1s)	195s	218s
MfgDyn (DT=5s)	42s	47s
MfgDyn (DT=30s)	10s	12s
MfgDyn (DT=5min)	6s	6s

(1) Software used by company Z.

(2) Model built by the company for this experiment.

(3) Existing model of the company for the 'b' model, including all information. (distributions in cycle times, and a large database with breakdowns occurrences and maintenance times)

Table 41: Model execution time (Case Z)

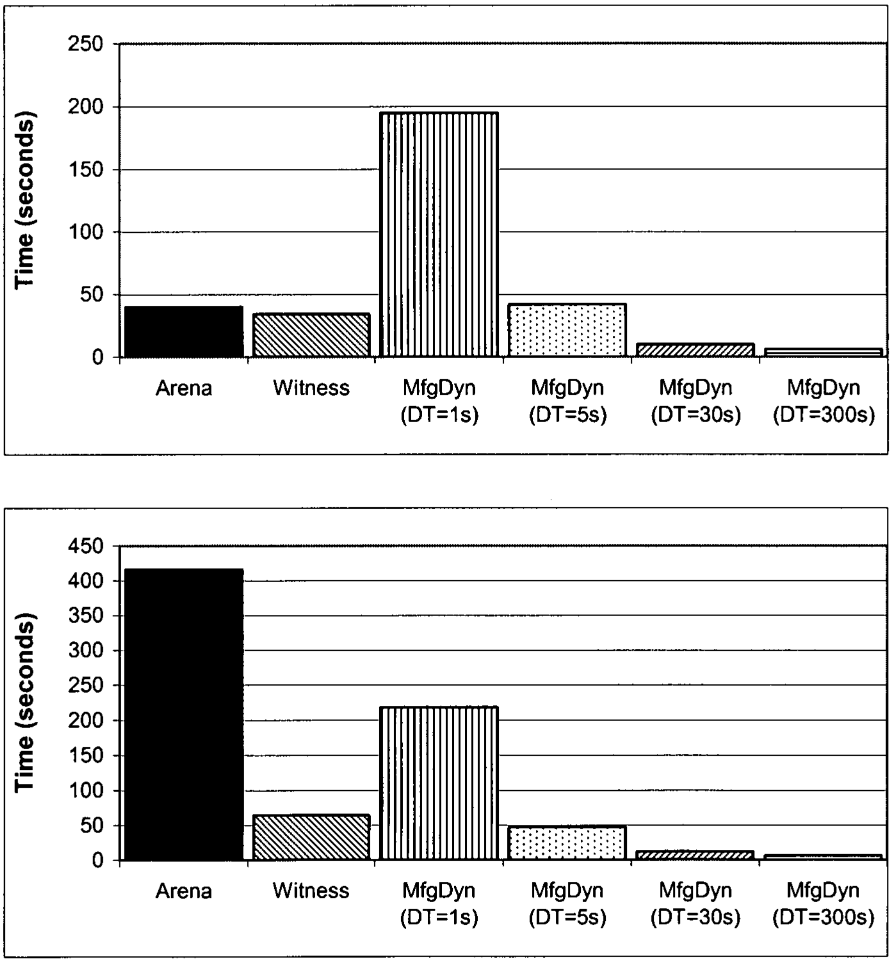


Figure 64: Model execution time: a) Model A, b) Model B (Case Z)

6.4 Analysis of results and key findings

Section 6.2 presented the case study design, where four performance assessment criteria were selected (see Section 6.2.1). Then, Section 6.3 compiled the information involved in the three cases, and presented a summary of the results. This section analyses and discusses these results. The analysis, in addition to the literature and the information gathered in the first stage of the research, will provide an improved understanding about the characteristics and limitations of the SD technique when applied to manufacturing systems. The structure of this chapter is based on the four performance measures previously selected, namely: (i) Accuracy of the technique, (ii) Modelling time, (iii) Execution time and (iv) Required skills.

6.4.1 Accuracy of the SD technique

The results obtained from the cases show that whilst accuracy of DES based models depend only on the quantity of data and its accuracy, the accuracy of SD based models depend also on the selection of an appropriate DT. In addition, the characteristics of the SD technique (see Section 2.2.2) have an effect on its suitability for the manufacturing systems design context. For example, the assumption that material is treated as a flow, and thus can be processed in a continuous way, limits the application of the SD technique when the simulation time horizon is reduced.

Case X supports this limitation. As can be seen in Table 36, when a short time horizon is simulated (1, 2, and 5 minutes) the result of the SD model is strongly dependent on the value chosen for DT, and differences of more than 67% were found between the results produced by MfgDyn and Witness. The reasoning for this finding is as follow: if DT is small, the material will flow through the system quicker (but in smaller quantity) and therefore, the first product will be manufactured faster. On the other hand, if the value of DT is high, the system will require longer time to process the first product.

All models analysed in the three case studies are linear. This is to reduce the number of causes that can modify the value of the outputs, in order to focus on individual causes and thus, find out the causes of success or failure of SD. In this context, SD was found to be very accurate (as compared to DES) when modelling linear flows (when no detailed data about interferences such as breakdowns or setups exist) (Case X). In addition, accuracy of results of SD models was found to increase when the time horizon increases, obtaining differences less than 1% from Witness models (see Table 36 and Table 37).

When several products are manufactured in batches (see Case Y), with predefined priorities, SD was found to manage the priorities appropriately; however, the continuous use of batches, made MfgDyn produce the predefined quantity of products (4096) faster than Witness. While DES based models have to wait until the full batch size is available, SD based models can decompose them and therefore, reduce the delays associated with these scenarios. Thus, it seems apparent that SD is less suited than DES when products are manufactured in large batches.

Breakdowns in the form of distribution were also modelled in Case Y. However, due to the short time horizon simulated (and the high MTBF) no breakdowns occurred in the Witness model during the simulation. However, it is obvious that this occurrence would have increased the gap between the results of both models.

In general, results show that DES supports credibility better than SD. This distinction has mainly arisen because DES enables the construction of a model to include more detail than SD. For example, a DES model can represent individual products in a queue before a machine, whereas a SD model will be limited to showing an accumulation of product flow. As stated by Baines (1994), higher credibility can be considered to be roughly proportional to an increase of model detail.

6.4.2 Modelling time

Modelling time and accuracy present a dilemma (see Section 2.4); whether to choose an approach that provides faster model building rate, but to a lower level of accuracy, or a considerably slower model building rate, but eventually a better value of accuracy. A faster model building rate will mean that alternative manufacturing designs can be evaluated in less time. In this case, MfgDyn (the tool developed in this research) was demonstrated to be successful in practice. While many authors criticise the notation of SD when applying to the manufacturing context, the development of MfgDyn has demonstrated that if a tailored tool is provided, modelling time can be as good (and sometimes better) than equivalent models built in DES tools (see Case X: Table 35 and Case Y: Table 39). Users of MfgDyn found useful the intuitive interface and the way in which the flows are created, by simply clicking in the two elements that need to be linked. It must be noted, however, that one of the reasons why MfgDyn is more intuitive and simple to use than Witness, is due to the lower number of features included in it.

DT can be a disadvantage for the application of SD in industry. Section 6.4.1 has mentioned some of its effects on the accuracy of the results. In addition, as stated by practitioners in first stage of this research, the survey conducted revealed that simulation practitioners do not feel comfortable with this parameter. The main reason for this is that the selection of an optimum value of DT can be time consuming. Experiments must be repeated often to find out if the DT value provides the level of accuracy required without compromising the speed of execution. Thus, it looks apparent that the SD technique can be benefited if an algorithm is created to search for an optimum DT value without user intervention.

6.4.3 Execution time

Case X supported literature (see Section 2.4.2) by demonstrating that SD models can be considerably faster than DES models under certain circumstances (see Figure 58). The model developed in Case X consisted of a linear flow where stoppage causes (breakdowns and setups) were considered as averages in both the SD and DES tools. In this context, SD consistently required less time to run a model, by taking approximately 12% (when DT=15s) of the time taken for DES for a similar level of accuracy. Reasons for this advantage on the speed were identified in the literature, and include aspects such as:

- ➢ In certain scenarios, such as Case X, SD models require much less calculations than DES for a similar accuracy.
- ➢ The management of event lists by DES slows down the simulation speed.

In addition, Case Z demonstrated that (as stated by Sterman (2000) - see Section 1.1), execution speed of SD models evolve in a better manner than the DES ones when the size of the model increases. In this case study (Case Z), MfgDyn increased its execution time between 10 and 20% when the model increased from one to three machines. In the same scenario, the model developed by Witness had to increase its execution time by 88% for simulating the new conditions. However, the gap between the execution time required by SD and DES based models can increase substantially if detailed information is included in the DES models. An example is provided in Table 41.

However, there are a number of scenarios where the advantage in terms of execution speed of SD models is reduced, eliminated or even overcome. Case Y exhibits a scenario where SD can be slower than DES. If a system is processed in batches, DES based tools consider the batch as a product, and therefore, the number of events is reduced according to the size of the batch. In this case, the model constructed in Witness was found to be considerably faster than the model constructed with SD. A solution to this problem would consist of increasing the value of DT (to reduce also the number of times that the system is checked and calculated) but then, the accuracy problem described in 6.4.1 would be increased.

Another example of scenarios that are not suited to be modelled by SD consists of systems where a sequence of events takes place in a short period of time, followed by a long period of inactivity. The main reason for this is the selection of the DT value. If a small DT value is selected, the system will be able to detect the sequence of events, but will be slow because it will stop the system unnecessarily during the long periods of inactivity. On the other hand, if a high value of DT is selected, the simulation will be fast, but events that take place in a short period of time will not be detected.

6.4.4 Required skills

The developed tool (MfgDyn) was found to be intuitive and required relatively low skills. Participants of the case study were already familiar with the development of models using DES, and became familiar with MfgDyn after a few hours. Although it seems apparent that there is a relationship between the skills required and the number of features included in a tool, quantitative data has not been gathered to determine the extent to which the developed tool is easier to use than existing DES and SD based tools.

As stated earlier, a reason for the simplicity of use (and low skill requirements) can be the number of features implemented on the tool. While commercial simulation tools include a large variety of functions, the developed tool is still in a prototype stage and its functionality is limited. If further development is conducted, aspects such as the inclusion of proprietary programming languages to model more complicated control logic must be considered.

6.5 Summary

This chapter has described the method, process and results of the three case studies conducted in the third stage of this research. The chapter started by explaining the methodology followed in this stage, followed by the design of the case study. This design involved aspects such as: (i) Performance assessment criteria, (ii) Selection of the industrial cases, (iii) Role of the participants.

The second part of this chapter compiles and shows the main characteristics of the processes analysed, as well as the argument for their selection. It also provides a summary of the information gathered and developed both in the text and graphical format.

Finally, the data obtained from the case studies is discussed. The analysis of this data has focused on answering the third objective of the research aim: *"Assess the true capabilities of SD, assessing performance against a typical DES modelling capability"*. Four performance criteria were selected, and a discussion about the findings obtained from the case studies and their implications in the usability of SD in the manufacturing systems design context was conducted. In this way, this chapter has improved the understanding about the characteristics and limitations of the SD technique when applied to a number of typical manufacturing systems scenarios.

Chapter 7

CONCLUSIONS AND FUTURE WORK

This chapter discusses and concludes the findings of the research documented in this thesis. The chapter discusses how the research aim and objectives have been met, the limitations of the findings and the further opportunities for research arising from this work.

7.1 Discussion of the research aim and objectives

When considering simulation of manufacturing systems, the literature review (see Section 2.4) identified gaps in the knowledge that required further research. The gaps led to the development of a research aim, which was to:

> *Determine the extent to which System Dynamics can provide a credible alternative to Discrete Event Simulation in the process of manufacturing system design.*

The research aim was addressed by completing the following objectives:

1. Understand what is needed of a System Dynamics (SD) modelling tool for it to complement the needs of a manufacturing system designer.
2. Represent the capabilities of SD in a modelling tool tailored to manufacturing system design.
3. Assess the true capabilities of SD, by applying the modelling tool to real manufacturing problems, and assessing performance against a typical Discrete Event Simulation (DES) modelling capability.

The following discussion will highlight how the research aim and each of the three objectives identified above have been addressed.

7.1.1 Objective 1: Business needs

Objective 1 was to understand what is needed of a SD modelling tool for it to complement the needs of a manufacturing system designer.

A review of published literature (see Chapter 2) provided a set of common definitions for modelling and simulation related to the manufacturing context (see Section 2.1.1), in order to understand what industry understands when these definitions are used. Following this, different approaches to classify modelling techniques were reviewed, and the classification provided by Baines (1994) was adopted (see Section 2.1.2). This classification considered both DES and SD techniques as simulation techniques; a sub-category of analytical techniques. A review of the modelling techniques described in Section 2.1.2 identified the main particularities of each technique and found that simulation techniques are most likely able to deal with the modelling of complex manufacturing systems because; while techniques such as mathematical modelling rely on *"strictly mathematical equations"*, the simulation approach implies the use of statistical estimation (Rubinstein and Melamed, 1998; cited in Section 2.1.4), and therefore the usability and flexibility of simulation techniques is expanded.

A review of the work conducted by other researchers found that both DES and SD simulation techniques were developed in the 1950s (see Section 2.1.4.1 and Section 2.1.4.2) and have evolved continuously, and are considered to aid practitioners in a number of scenarios. However, this review also revealed that DES is the preferred simulation technique of most practitioners in the manufacturing systems design field (see Section 2.3); the examples of applications where SD is applied to this task being very rare (Coyle, 1995). This situation has resulted in an extensive offer of DES tools tailored to model manufacturing systems, whereas the offer of SD tools tailored for this task is non-existent (see Section 2.4.3); thus, the gap between the utilisation of both techniques is increasing.

The mechanisms of both DES and SD technique were detailed in Section 2.2 to identify if the SD technique can successfully be used as an alternative to DES. This review described the main elements used in each of the above mentioned techniques, as well as the time control methods associated with each technique. It also revealed that DES uses events (both scheduled and conditionals) to control the time (Kay, 1994), whereas SD uses a fixed time interval, also called Delta Time (DT). While manufacturing systems' time management seems to be closer to the DES technique, the literature revealed that this type of control can be very time consuming if many events occur in a short period of time (Aitchison, 1995). On the other hand, the accuracy and execution time of SD models was found to be dependent on the resolution of the model (defined by DT) and, following the suggestions of Zaraza (1998), a method for selecting appropriate values of DT was provided. The literature also revealed that DES based tools are closely related with the simulation of stochastic systems (Law and Kelton, 1991), usually providing a library of standard distributions ready to use, whereas pure SD models do not use stochastics.

The literature then highlighted the importance of simulation in the manufacturing system design process (see Section 2.3), while common performance measures obtained in practice were identified (Okudan and Kabadayi, 2001). In addition, an overview of the design process following the suggestions of Slack *et al.* (1999) was presented, and the time saved in making decisions earlier in the design process were highlighted. As stated by Moore (1999), if manufacturing system design can be aided during the early stages with the appropriate tools, modifications in the design can be reduced, and consequently money and time can be saved. Thus, an aggregate level simulation might fit in this context.

Section 2.2 of the literature also described the original notations of both DES and SD techniques; named 'activity cycle diagrams (Pidd, 1988)' and 'level and rates diagrams (Sterman, 2000)', respectively. It found both notations useful in a large variety of systems (see Section 2.2.1.2 and Section 2.2.2.3). However, the literature also revealed that these forms of notation are not suited to complex systems, since they can make the model 'unreadable'. In addition, Section 4.3 revealed that the evolution of DES and SD tools has been different; while DES tools now include iconic model representation (including predefined elements with a set of attributes), SD tools have evolved from equations to a graphical notation, but maintain the 'level and rate' convention (Pidd, 1988), being less accepted by practitioners in the manufacturing systems field. Therefore, a possible reason for the misuse of SD in this task (Hlupic, 2000) was identified.

The needs for a simulation based on the SD principles cannot rely only on literature. Thus, interviews (based on a semi-structured technique) were conducted to validate the findings described above, and gather user preferences. This was addressed by carrying out interviews at 14 organisations from three different sectors (manufacturing, academia, software development) (see Section 4.2 and Appendix D). This sample number was considered adequate, providing a greater understanding about the practice of manufacturing in industry, and their needs.

The quantitative analysis carried out for the 'closed' questions of the survey revealed a set of features that are essential in a tool tailored for manufacturing systems design. Examples of these are the inclusion of 'parts', 'machines', 'buffers', performance measures such as 'throughput', 'machine rate', etc. However, differences arose when interviewees were asked about SD mechanisms that are not currently included in DES tools. Examples of these are the continuous nature of 'parts' (material is treated as flows) and the inclusion of DT as a time control method. While members from academia were familiar with these terms and considered them *"the essence of SD"* and therefore, for implementation, practitioners had the opposite view. In addition, the open discussion following closed questions permitted the researcher to gather information about the use of simulation tools in practice (for example, the mouse is the preferred method to construct models) that helped to define specifications not covered previously.

7.1.2 Objective 2: SD based tool tailored to manufacturing system design

Objective 2 of this research was to represent the capabilities of SD in a modelling tool tailored to manufacturing system design.

The second objective was addressed by compiling information from the following: (i) literature, (ii) interviews, (iii) analysis of existent DES and SD commercial tools (see Section 5.1). A list of key inputs provided by the sources described above was listed (see Section 5.2), and their consistency amongst the different sources was checked. In this screening stage, some desirable features were discarded. For example, although the 'lead time' performance measure was moderately appreciated by practitioners, it was decided not to include it in the tool due to its inconsistency with the SD technique. A second screening stage consisted on checking the consistency of the previously developed requirements with the research scope. For example, desirable features, such as 'real time animations' were discarded due to the time required for their implementation and the low relation between this feature and the aim of the thesis.

The requirements specification led to the design stage, where requirements were then translated into working parameters to define 'how' the desired features need to be implemented in the tool. Following the guidelines provided by the authors in this field (see Section 5.1.2.2) the design of the tool was divided in two tasks, namely: (i) User interface design and (ii) design of analytical capabilities.

The user interface was designed using a Multi Document Interface (MDI), where five child forms were allocated. This design was conducted to allow inexperienced users to follow the typical modelling project stages (McHaney, 1991) and improve usability. In addition aspects related to developing user-friendly interfaces were considered to reduce the learning curve.

The analytical capabilities of the tool included the selection of attributes for the elements to be integrated in the tool, and the algorithms implemented to obtain the selected performance measures. Thus, a set of attributes for each element (parts, machines and buffers) was developed and its operation was described. Obtaining performance measures in a continuous technique involved the use of numerical integration methods. The different numerical integration methods available, such as 'rectangular', 'trapezoidal', etc. were identified (Wang and Skeel, 2003), and the 'rectangular' method was selected for implementation in the tool. This decision was based in the lower calculation time required by this method with respect to the other methods considered. Then, the mechanisms for calculating the selected performance measures using this type of graphical integration were provided.

The final stage for achieving the second objective involved implementation of the tool, using Visual Basic as the programming language (see Section 5.1). This task was addressed following the suggestions provided by Petroutsos and Hugh (1999); dividing the tool into small modules (see Section 5.4). Benefits of this practice were identified, including reusability and reduced testing time. All modules were then integrated and validation tests were conducted.

Validation included the testing of all individual modules developed, their integration into the final application, and the analysis of the performance measures. Considering that one of the objectives of this research involved the benchmarking of the developed tool with a commercial DES tool in terms of accuracy, building time and execution time, two tests were conducted to validate the accuracy and execution time of the tool. The execution time test was conducted by comparing the time required by the tool to simulate a test model with the time required by a commercial SD based tool to do the same task under the same conditions. After executing the test several times, this tool was found to be 5.88% slower than the commercial one. Possible reasons for this included the use of a high level programming language, programme flows or programming skills.

7.1.3 Objective 3: Assess the true capabilities of SD and assess its performance against a typical DES tool

Objective 3 of this research was to assess the true capabilities of SD, by applying the modelling tool to real manufacturing problems, and assessing performance against a typical DES modelling capability.

This objective was addressed by conducting three case studies using companies as test-beds, and modelling selected processes with a commercial DES based tool (Witness) and the tool developed in the stage two of this research (MfgDyn). The selection of the number of case studies to be conducted was justified after following the suggestion of Yin (1994) and considering the nature of this research. In addition, the limitations of the developed tool were considered in order to select manufacturing systems that can be modelled within the tool. In addition, simple processes were selected, in order to isolate the different factors that have effect on the accuracy and speed of execution of SD models.

The findings presented in this stage suggest that the SD tool is able to benefit companies in their modelling task, especially when aggregate simulation is required (see Section 6.4). In addition, the case studies conducted revealed that the accuracy obtained by SD can be very similar to DES when long running periods are simulated with no significant stoppages (such as breakdowns or setups). In addition, when simulating this type of process, SD is definitely faster than DES. An exception for this rule appears when simulating systems with transfer batches. Although in this case the accuracy of the SD model is comparable to the DES one, the advantage in the running time can be minimal or even negative. On the other hand, SD seems to be less suited to industry when an analysis of instantaneous moments of time is required, for example in systems where breakdowns or setups are predominant. A summary of the findings obtained from the cases studies (see Section 6.4) is provided in Table 42.

SD vs. DES	
SD better than DES when	**DES better than SD when**
Long running time (horizon) Models with large number of elements Limited resources (data, skills, money) Rough analysis (average results)	Transition periods Transfer batches Breakdowns and setups Complex control policies "Irregular" events Instantaneous analysis

Table 42: Summary of key findings

In addition, some limitations of its application in terms of: (i) Accuracy, (i) Speed and (iii) Suitability were also reported. Examples of these are the limitations provided by the SD technique to model breakdowns or any disruption in general, as well as its inadequacy for obtaining accurate 'instantaneous' results.

As stated in Section 2.3, the inclusion of simulation techniques earlier in the design process can benefit industry by reducing both the development time and cost. Thus, the companies most likely to obtain the greatest benefits from using the developed tool are the ones that:

- Require aggregate modelling.
- Require effective results (where high accuracy is not a requirement).
- Small companies with simple processes.
- Companies that execute long time horizon simulations.
- Companies that want to introduce simulation.

However, as explained later in this chapter, the tool requires further development and inclusion of more features if it is to be truly compared with a DES tool in a broader number of scenarios.

Although this tool has been tailored to manufacturing system design, the concepts of SD can be extrapolated to other fields. An example of a possible new application is a call centre. In this example, the same concepts described in this thesis can be applied by simply translating parts by calls, machines by operators, and stocks by switchboards.

7.2 Contribution

This research has made a primary contribution to the body of knowledge; that is: To provide a much improved understanding of the capabilities of SD as an aid to manufacturing systems design.

This contribution is supported by the following arguments: First, published literature assumes that DES is the preferred technique applied in the manufacturing system design process, while techniques such as SD are very rarely mentioned in this context. However, no empirical analysis exists that quantifies the validity of SD when applied to this task. Moreover, literature tends to ignore SD in this context, assuming that the application of simulation in the manufacturing systems design is restricted to DES. This research analysed the mechanisms of SD, and found that some characteristics of this technique can benefit the application of simulation in industry. This concept was taken forward and a tool was developed that considered literature and user needs. The benchmarking of the developed tool (based on the SD principles) and a commercial tool (based on the DES principles) found that SD is sufficiently accurate when modelling systems under stable conditions, whereas transition periods are better suited for DES, due to the nature of this technique. In addition, the case studies demonstrated quantitatively that model execution time can be significantly faster in SD (than in DES) for an equivalent system and level of accuracy. However, this advantage of SD is limited to the characteristics of the model.

Second, the researcher has found that the user interface appears to have a significant impact on the lack of adoption of SD techniques within the manufacturing industry. The lack of tailored elements to conduct simulation within this context makes it difficult to construct the models. Although the tool developed is in a prototype stage, and only some features are implemented, model building time has been substantially improved and the learning curve reduced. In addition, the survey conducted in the first stage of the research revealed that practitioners strongly favour Object-Oriented and iconic representation of system elements. Thus, it is apparent that if SD wants to be included successfully within the manufacturing system design context, a tailored tool is required for this task.

7.3 Limitations

As in any research, this work also has a number of limitations. The primary limitations of this research concern the development of the tool, and its evaluation through case studies.

The first limitation concerns the development of the tool. In order to increase practicality, the comparison between the SD technique and the DES one was done through gathering user needs and the development of a computer tool. The limitations involved with this methodology include the number of features that can be implemented by the researcher through the PhD program. If a theoretical approach to compare the capabilities of DES and SD were conducted, the number of compared features would have possibly been higher. However, if the comparison between the DES technique and SD is done using existent software, the extent to which the user interface is responsible for the misuse of SD cannot be compared in practice. Thus, in order to compare both technical characteristics and usability in real life scenarios, the diversity of comparisons were reduced.

The second limitation is related to the evaluation through case studies. The tool developed in this research has been tested using real cases. However, only the researcher and a group of students of Mondragon University have used the tool. The researcher would have preferred to model the case studies by members of the companies involved in the case studies; in order to avoid bias, and considering the time constraints, it was not possible to do so.

Sample size is also important when comparing the capabilities of this technique. Although this work has been based around three case studies and fourteen interviews, the findings obtained in the research cannot be considered absolutes, and the performance measures obtained have to be considered as approximate values. Thus, further work in this area would be required if confidence is to be increased.

In addition, other limitations of this research are described below.

- The literature showed that very few people are working with the application of SD to manufacturing systems design and, so validation of findings with peers was not possible.
- Although the developed tool is based on the SD principles, some 'discrete' elements have been included to increase the applicability of the tool. Thus, as stated in Chapter 5, the rate of the machine is controlled in a 'semi-discrete' manner, and priorities are implemented in a 'discrete' manner.
- In the manufacturing context, simulation practitioners have incorrect perception about the true capabilities of SD. A generalised opinion consists of thinking that if DES tools can do everything, then SD is not needed. However, in a final commercial tool, most mechanisms inherent to the SD technique could be masked within the capabilities of the tool
- This research created some simple elements that were used to model manufacturing systems. Thus, SD was used mechanically, and therefore, the full potential of the SD technique was not used. An example of the mechanical application of SD through the developed tool can be found in the modelling of machines. These models are designed to work according to a set of predefined rules, with first order control methods. Although this can reduce the skills required by final users, it must be considered as a limiting factor to the complete application of the SD technique.

7.4 Recommendations for further research

The research has outlined the comparison, in real life situations, of the two analytical simulation techniques, namely DES and SD. It would benefit from further research into the following areas:

- Further development of the tool developed during this research. If new and improved features are included into the tool, its usability, and therefore acceptance, will be increased.
- Development of a library of generic manufacturing elements within commercial SD tools. This would allow practitioners to use existent SD tools (usually cheaper than equivalent DES tools) but tailored to the manufacturing systems design process.
- Development of a library of continuous elements within commercial DES tools that are executed following the principles of SD. Considering the vast majority of practitioners that use simulation to model manufacturing process use tools based on the DES technique, the inclusion of these continuous elements within the existent tools will expand their usability.
- Develop a truly balanced and integrated tool based on the principles of both the SD and DES technique. Analysis of existent software revealed that the latest trends include the inclusion of continuous elements in DES models and vice versa. In the researcher's opinion, combining SD elements with DES can, in some instances, prove valuable because it can combine the strengths of both techniques. This might be especially useful because it enables rough cut modelling earlier in the design process, and the possibility of adding more detail into the model to conduct detailed simulation later in the design process.
- Conduct a survey to gain a broader understanding of the impact that SD can have on industry; this would help to identify specific research areas. This work did not set-out to analyse the implications of SD in a wide number of contexts and sectors, but a survey could prove beneficial.

Chapter 8

REFERENCES

Ackoff, R.L. and Sasieni, M.W. (1968). *Fundamentals of Operations Research.* John Wiley and Sons, New York, USA.

Adesola, M.O. (2003). *Development and assessment of a methodology for business process improvement.* PhD thesis, Cranfield University, Cranfield, UK.

Aitchison, K.J. (1995). *Rapid and accurate modelling of engine manufacturing systems.* PhD thesis, Cranfield University, Cranfield, UK.

Akins, C. (2002). *Introducing evaluation* (WWW document). http://people.cs.tamu.edu/cja3191/chapter10.htm. (Accessed 19th May 2003).

Albin, S. (1996). Generic structures: first-order negative feedback. *MIT working papers.* D-4475, Massachusetts, USA.

Albin, S. and Choudhari, M. (1996). Generic structures: first-order positive feedbacks. *MIT working papers.* D-4474, Massachusetts, USA.

Alfeld, L.E. (1994). *The System Dynamics modelling methodology.* (WWW document). http://www.decisiondynamics.com/SystemDynamics/Modelling%20Method.htm . (Accessed 19th May 2003).

Allen, A.O. (1990). *Probability, statistics, and queuing theory with computer science applications,* 2nd ed. Academic Press, New York, USA.

Allen, R. (2002). Mathematical modelling / Computational science. In: *AiS Challenge: STI 2002.* June, 1-15 2002, Farmington, New Mexico, USA.

Alreck, P.L. and Settle, R.B. (1985). *The survey research handbook.* Irwin, Homewood, Illinois, USA.

Askin, R.G. and Standridge, C.R. (1993). *Modelling and analysis of manufacturing systems.* John Wiley and Sons, New York, USA.

Au, G., and Paul, R.J. (1996). Visual interactive modelling: a pictorial simulation specification system. *European Journal of Operational Research,* 91, 14-26.

Avni, T. (1999). Simulation modelling primer. *IIE Solutions*, 31(9), 38-41.

Baines, T.S. (1994). *Modelling in the evaluation of a manufacturing strategy.* PhD thesis. Cranfield University. Cranfield, UK.

Baines, T.S., Harrison, D.K., Kay, J.M., and Hamblin, D.J. (1998). A consideration of modelling techniques that can be used to evaluate manufacturing strategies. *The International Journal of Advanced Manufacturing Technology,* 14, 369-375.

Baines, T.S., and Harrison, D.K. (1999). An opportunity for system dynamics in manufacturing system modelling. *Production planning and control,* 10(6), 542-552.

Baines, T.S., and Kay, J.M. (2002). Human performance modelling as an aid in the process of manufacturing system design: a pilot study. *International Journal of Production Research,* 40(10), 2321-2334.

Balbo, G., Bruell, S.C., and Ghanta, S. (1988). Combining queuing networks and generalized stochastic Petri Nets for the solution of complex models of system behaviour. *IEEE transactions on computers,* 37(10), 1251-1268.

Banerjee, S. and Basu, A. (1993). Model type selection in an integrated DSS environment. *Decision support systems,* 9, 75-89.

Banks, J. (1998). The future of simulation software: a panel discussion. In: *Proceedings of the 1998 Winter Simulation Conference*, Washington DC, USA, December 13-16, 1998, 1681-1687. ACM.

Banks, J. (2000a). Introduction to simulation. In: *Proceedings of the 2000 Winter Simulation Conference,* Orland, USA, December 10-13 2000, 9-16. ACM.

Banks, J. (2000b). Simulation in the future. In: *Proceedings of the 2000 Winter Simulation Conference,* Orland, USA, December 10-13 2000, 9-16. ACM.

Banks, J., Carson, J.S., and Nelson, B.L. (1996). *Discrete-Event system simulation.* 2nd ed, Prentice Hall, New Jersey, USA.

Barton, P.M. (1992). *The modelling and simulation of combined discrete/continuous processes.* PhD thesis. University of London, London, UK.

Barton, P.M. and Tobias, A.M. (1998). Accurate estimation of performance measures for system dynamics models. *System Dynamics Review,* 14(1), 85-94.

Baudoin, C. (1995). *Object-oriented manufacturing systems, a white paper.* (WWW document). http://www.omg.org/archives/mfg/msg00166.htm. (Accessed 9th January 2003).

Baumol, W.J. and Wolff, E.N. (1994). A key role for Input-Output analysis in policy design. *Regional Science and Urban Economics*, 24, 93-113.

Baxter, M.R. (1995). *Product design: A practical guide to systematic methods of new product development.* Chapman and Hall, London, UK.

Bellinger, G. (2002). *Mental model musings.* (WWW document). http:// www.outsights.com/systems/welcome.htm. (Accessed 19th May 2003).

Bender, E. (2000). *Introduction to mathematical modelling.* Dover Publications, UK.

Bhamra, R.S. (2003). *The nature of competence and resource-based concepts within UK manufacturing SME's: an exploratory study and framework.* PhD thesis. Cranfield University, Cranfield, UK.

Borland Software Corporation. (2003). *Delphi* [CD-ROM], ver. 7, [Computer programme]. Available distributor: 100 Enterprise way, Scotts walley, CA 95066, USA.

Boucher, A. (1995). Systems modelling in economics: use of object-oriented software. *Computers in Higher Education Economics Review,* 9(2), 12.

Boyer, K.K., and McDermott, C. (1999). Strategic consensus in operations strategy. *Journal of Operations Management*, 17, 289-305.

Brandimarte, P., and Vila, A. (1999). *Modelling manufacturing systems: from aggregate planning to real-time control.* Springer, Berlin, Germany.

Bready, R. (2000). *Microsoft bookshelf,* Microsoft, Seattle, USA.

Brenner, M., Brown, J., and Canter, D. (1985). *The research interview: uses and approaches.* Academic Press, New York, USA.

Brinberg, D. and McGrath, J.E. (1985). *Validating and the research process.* SAGE, London, UK.

Brooks, F.P. (1975). *The mythical man month,* Addison-Wesley, Reading, Massachusetts, USA. In: Zelkowitz, M.V., Shaw, A.C., and Gannon, J.D. (1979). *Principles of software engineering and design.* Prentice-Hall, Englewood Cliffs, New Jersey, USA

Burns, R. (2000). *Introduction to research methods.* SAGE, London, UK.

Carrie, A. (1988). *Simulation of manufacturing systems.* John Wiley and Sons, Glasgow, UK.

Casti, J.L. (1997). *Would-be worlds: how simulation is changing the frontiers of science.* John Wiley and Sons, New York, USA.

Centeno, M.A., and Carrillo, M. (2001). Challenges of introducing simulation as a decision making tool. In: *Proceedings of the 2001 Winter Simulation Conference,* USA, December 8-11 2000, 17-21. ACM.

Chan, F.T.S. (1995). Using simulation to predict system performance: a case study of an electro-phoretic deposition plant. *International Journal of Integrated Manufacturing Systems,* 6(5), 27-38.

Chan, F.T.S., and Jiang, B. (1999). Simulation-aided design of production and assembly cells in an automotive company. *Integrated Manufacturing Systems,* 10(5), 276-283.

Chang, E.C., and Wong, S.M.L. (2002). *Corporate governance, political interference and corporate performance of China's listed companies.* Hong Kong Institute of Economics and Business Strategy Working Paper N1037, Hong Kong.

Chawla, S.K., Balakrishnan, P.V., and Smith, M.F. (1992). Mail response rates from distributors. *International Marketing Management,* 21, 307-310.

Chawla, S.K., and Nataraajan, R. (1994). Does the name of the sender affect industrial mail response? *Industrial Marketing Management,* 23, 111-115.

Checkland, P.B. (1981). *Systems thinking, systems practice.* John Willey and Sons, Chichester, UK.

Checkland, P.B. (2000). Soft Systems Methodology: a thirty year retrospective. *Systems Research and Behavioural Science.* 17, 11-58.

Churchill, G.A. (1998). *Marketing research: methodological foundations.* 7th ed. The Dryden Press, London, UK.

Chwif, L., Paul, R.J., Ribeiro, M. (1999). Combining the best of the two: an activity cycle diagram / condition specification approach. In: *Proceedings of the United Kingdom Simulation Society Conference UKSIM'99.* Cambridge, UK, 7-9 Apr 1999.

Collins, R.S. and Cordon, C. (1997). Survey methodology issues in manufacturing strategy and practice research. *International Journal of Operations and Production Management,* 17(7), 697-706.

Colquhoun, G.J., Gamble, J.D., and Baines, R.W. (1989). The use of IDEFo to link design and manufacture in a CIM environment. *International Journal of Operations and Production Management.* 9(4), 48-65.

Compton, D. (1988). *Design and analysis of integrated manufacturing systems.* National Academy Press, Washington D.C., USA.

Connell, L.M. (2000). *Categories, concepts and co-occurrence: modelling categorisation effects with LSA.* MSc thesis. University of Edinburgh, Edinburgh, UK.

Conway, R. (1978). *A primer on disciplined programming,* Winthrop Publishers, Cambridge Massachusetts, USA.

Coronado, A.E. (1996). Mental simulation: adding constant flows. *MIT working papers.* D-4546. Massachusetts, USA.

Coyle, R.G. (1977). *Management system dynamics.* John Willey and Sons, Chichester, UK.

Coyle, R.G. (1985). Representing discrete events in system dynamics models. *Journal of Operational Research Society,* 36(4), 307-318.

Coyle, R.G. (1996). *System Dynamics modelling: a practical approach.* Chapman and hall, London, UK.

Coyle, R.G., and Exelby, D. (2000). The validation of commercial system dynamics models. *System Dynamics Review,* 16(1), 27-41.

Cugola, G., and Ghezzi, C. (1998). Software process: a retrospective and a path to the future. *Software process, improvement and practice,* 4, 101-123.

De Geus, A.P. (1988). Planning as learning. *Harvard Business Review.* March-April.

Delamont, S. (1992). *Fieldwork in educational settings: methods, pitfalls and perspectives,* Falmer, London, UK.

De Marco, T. (1978). *Structured analysis and system specification.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA.

Denzin, N.K. (1988). *The research act: a theoretical introduction to sociological methods, 3rd ed,* Prentice-Hall, New Jersey, USA.

Denzin, N.K. and Lincoln, Y.S. (2000). The discipline and practice of qualitative research. In Denzin, N.K. and Lincoln, Y.S. (Eds.). *Handbook of qualitative research, 2nd ed.* Prentice Hall, New York, USA. 1-28.

Department of Trade and Industry (DTI). (2002). Lord Sainsbury gives double boost to North East manufacturing. *May 2002 press office.* UK.

Diaz, R. (2003). *Teoria de colas y simulacion de eventos discretos.* Prentice-Hall, New York, USA.

Dijkstra, W. and Van der Zouwen, J. (1982). *Response behaviour in the survey interview.* Academic Press, New York, USA.

Donzelli, P., and Iaezolla, G. (2001). A hybrid software process simulation model. *Software process improvement and practice,* 6, 97-109.

Drever, E. (1995). *Using semi-structured interviews in small-scale research, a teacher's guide.* SCRE, Glasgow, UK.

Duncan, W.R. (2000). *A guide to the project management body of knowledge, 2000 ed.* PMI Publishing Division, Newtown Square, USA.

Dunham, W. (1999). *Euler: the master of us all.* Mathematical Association of America, USA.

Easteal, C., and Davies, G. (1989). *Software engineering: analysis and design.* McGraw-Hill, London, UK.

Edeler, H. and Krause, O. (1996). Integrated enterprise engineering: a case study. In: *Proceedings of the conference on agile and intelligent manufacturing systems.* Troy, New York, USA, October 1996.

Eisenhardt, K.M. (1989). Building theories from case study research. *Academy of Management Review.* October, 532-550.

Ellis, E.R. (1998). *System Dynamics, Computer-Based Tool, and Operational Concept Wargaming* (WWW document). http://206.239.103.141/edoweb/jmag/jmag.htm. (Accessed 19th May 2003)

Ellman, T. (2000). Software development methodology. Vassar College, New York, USA. (Unpublished work).

Ellson, T. (2002). *Improving adoption levels of manufacturing strategy formulation processes.* PhD thesis. Cranfield University, Cranfield, UK.

EUSTAT. (2002). Empresas y su empleo en la C.A. de Euskadi. http://www.eustat.es/elem/ele0002000/tbl0002023_c.html.

Federal Information Processing Standards Publication. (1993). *Standard for Integration Definition for function modelling (IDEFo).* (FED-STD-183), National Institute of Standards, USA.

Fink, A. (1998). *Conducting research literature reviews, from paper to the internet.* SAGE, London, UK.

Fishman, G.S. (1973). *Concepts and methods in discrete event digital simulation.* John Wiley and Sons, New York, USA.

Foddy, W. (1993). *Constructing questions for interviews and questionnaires: theory and practice in social research.* Cambridge University Press, Cambridge, UK.

Ford, D. (2001). *The next generation planning board: a visible solution for effective manufacturing planning and control for a process manufacturing environment.* PhD thesis. Cranfield University, Cranfield, UK.

Ford Motor Company. (2002). *2002 Corporate citizenship report* (WWW document). http://www.ford.com/en/company/about/corporateCitizenship/principlesProgress Performance/our-principles/environment-manage.htm. (Accessed 19th May 2003).

Forrester, J.W. (1961). *Industrial dynamics.* MIT Press, Cambridge, USA.

Forrester, J.W. (1969). *Urban dynamics.* Productivity Press, Portland, Oregon, USA.

Forrester, J.W. (1971). Counterintuitive Behaviour of Social Systems. *Technology review*, 73(3), 52-68.

Forrester, J.W. (1980). *Principles of Systems*, 2nd ed. MIT Press, Cambridge, Massachusetts, USA.

Forrester, J.W. (1994). System dynamics, systems thinking, and soft OR. *Systems Dynamics Review,* 10(2-3), 245-256.

Fowler, M. and Scott, K. (1999). *UML Distilled: a brief guide to the standard object modelling language*, 2nd ed. Addison Wesley, Massachusetts, USA.

Gallaher, M.P. and Martin, S.A. (1999). *Benefit analysis of IGBT power device simulation modelling.* Research Triangle Institute (RTI), North Carolina, USA. (Unpublished).

Gamma, E. (1995). *Design patterns.* Addison Wesley. Massachusetts, USA.

Gane, C. and Sarson, T. (1979). *Structured systems analysis: tools and techniques.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA.

Garrick, J. (1999). *Programming standards* (WWW document). http://www.citilink.com/~jgarrick/vbasic/standards/design.html (Accessed 19th May 1999).

Garrido, J.M. (2001). *Object-oriented discrete-event simulation with Java.* Kluwer Academic, New York, USA.

Garson, G.D. (2002). *Guide to writing empirical papers, theses and dissertations.* Marcel Dekker Inc., New York, USA.

Gershefski, G.W. (1970). *Corporate models, the state of the art,* University of Washington, Seattle, Wash. In: Shannon, R.E. (1975). *Systems simulation, the art and science.* Prentice Hall, London, UK.

Gilbert, N., and Troitzsch, K.G. (1998). *Simulation for the social scientist.* Open University Press, London, UK.

Gill, J. and Johnson, P. (1991). *Research methods for managers,* Chapman Publishing Ltd, London, UK.

Gillham, B. (2000a). *Developing a questionnaire.* Continuum, London, UK.

Gillham, B. (2000b). *Case study research methods.* Continuum, London, UK.

Gordon, G. (1978). *System simulation.* 2nd ed. Prentice Hall, New Jersey, USA.

Gray, P. (1984). Using the interactive financial planning (IFPS) for stochastic simulation. *Simulation,* December 1984, 286-292.

Green, P.E. and Tull, D.S. (1978). *Research for marketing decisions.* Prentice Hall, New Jersey, USA.

Greenfield, T. (1996). *Research methods: guidance for postgraduates,* Arnold, London, UK.

Groves, R.M. and Khan, R.L. (1979). *Surveys by telephone: a national comparison with personal interviews.* Academic Press, New York, USA.

Guasch, A., and Piera, M.A. (2001). Actividades de transferencia de tecnología en el ámbito de la simulación de procesos industriales. In: *Proceedings of the Red Temática de Simulación Digital y su aplicación a procesos industriales,* Valladolid, Spain, May 7-8 2001, Universidad de Valladolid.

Haider, S.W., et al. (1986). Simulation software products for analysing manufacturing systems. *Industrial engineering,* 18(7), 98-103.

Hall, R.W. (1990). *Queuing methods: for services and manufacturing.* Prentice hall, London, UK.

Halstead, M. (1977). *Elements of software science,* Elsevier North-Holland, New York, USA. In: Zelkowitz, M.V., Shaw, A.C., and Gannon, J.D. (1979). *Principles of software engineering and design.* Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Hamel, J. Dufour, S., Fortin, D. (1993).*Case study methods.* Sage Publications, Newbury Park, California, USA.

Harrell, C., Ghosh, B.K., and Bowden, R. (2000). *Simulation using ProModel.* McGraw Hill, New York, USA.

Heckel, P. (1991). *The elements of friendly software design,* 2nd ed. Sybex, San Francisco, USA.

Helliwell, J.F. (2002). *Globalization and well-being.* UCB Press, Vancouver, Toronto.

Heragu, S. (1997). *Facilities design.* Brooks/Cole, California, USA.

High Performance Systems. (2003). *Stella/iThink* [CD-ROM], ver. 8, [Computer programme]. Available distributor: High Performance Systems, Inc., 46 Centerra Parkway, Suite 200, Lebanon, NH 03766-1487, USA.

Hills, P.R. (1971). *HOCUS.* P-E Group. Egham, Surrey, UK. In: Pidd, M. (1988). *Computer simulation in management science,* 4th ed. John Wiley and Sons, Chichester, UK.

Hlupic, V. (1999a). Discrete-event simulation software: what the users want. *SIMULATION,* 73(6), 362-370.

Hlupic, V. (1999b). Guidelines for selection of manufacturing simulation software. *IIE Transactions,* 31, 21-29.

Hlupic, V. (2000). Simulation software: a survey of academic and industrial users. *International Journal of Simulation Systems, Science & Technology,* 1(1-2), 1-11.

Hlupic, V., and Paul, R.J. (1996). Methodological approach to manufacturing simulation software selection. *Computer Integrated Manufacturing Systems,* 9(1), 49-55.

Hlupic, V., Irani, Z., and Paul, R.J. (1999). Evaluation framework for simulation software. *International Journal of Advanced Manufacturing Technology,* 15, 366-382.

Hollocks, B.W. (1995). The impact of simulation in manufacturing decision making. *Control Engineering Practice,* 3(1), 105-112.

Holst, L., and Bolmsjo, G. (2001). Simulation integration in manufacturing system development: a study of Japanese industry. *Industrial management & data systems,* 101(7), 339-356.

Huang, J., et al. (2002). Performance evaluation of complex real-time systems: a case study. In: *Proceedings of the 3D progress workshop on embedded systems.* Jaarbeurs Utrecht, Netherlands, October 24 2002, 77-78.

Huff, B.L., Liles, D.H., Howell, B.J., Sanders, F.M., and Gaddis, B.P. (1991). The use of IDEF modelling techniques for small manufacturing enterprise modelling. In: *IDEF users group conference.* 153-169, Albuquerque, USA.

Ince, M. (2000). *Information systems: a methodology for capturing user requirements.* PhD thesis. Cranfield University, Cranfield, UK.

Johansson, H.J., McHugh, P., Pendlebury, A.J., and Wheeler, W.A. (1993). *Business Process Reengineering.* John Willey and Sons, Chichester, UK.

Kay, J.M. (1984). The use of modelling and simulation techniques in the design of manufacturing systems. *Proceedings of the IIE International Conference on Flexible Automation Systems,* July 1984, London, UK. IIE.

Keloharju, R. (1983). Relativity Dynamics. *Helsinki School of Economics*, Helsinki, Finland.

Kerlinger, F.N. (1986). *Foundations of behavioural research.* Holt Sanders International, Tokyo, Japan.

Keys, P. (1988). System dynamics, a methodological perspective. *Transactions of the Institute of Measurement and Control*, 10(4), 218-224.

Kheir, N.A. (1995). *Systems modelling and computer simulation,* 2nd ed. Marcel Dekker, New York, USA.

Khurana, A. (1999). Managing complex production processes. *Sloan management review*, 40(2), 85-97.

Kidder, L.H. (1981). *Research methods in social relations.* Holt, Rinehart and Winston. New York, USA.

Kiesmuller, G.P. (2003). A new approach for controlling a hybrid stochastic manufacturing / remanufacturing system with inventories and different leadtimes. *European Journal of Operational Research,* 147, 62-71.

King, C. (2001). *Nine steps to a successful simulation study* (WWW document). http://www.flexim.com/steps.html. (Accessed 19th May 2003).

Kinnear, T. and Taylor, J. (1996). *Marketing research: applied approach.* McGraw-Hill, New York, USA.

Kirkwood, C.W. (1998). *System dynamics methods: a quick introduction.* College of business, Arizona State University, USA.

Klemola, K., and Turunen, I. (2001). *State of mathematical modelling and simulation in the Finnish process industry, universities and research centres.* National Technology Agency, 107/2001. Helsinki, Finland.

Knuth, D.E. (1998). *The art of computer programming,* 2nd ed. Addison-Wesley. Massachusetts, USA.

Kover, A.J. (1983). The legitimacy of qualitative research. *Journal of advertising research,* January 1983, 49-50.

Krajewski, L.J., and Ritzman, L.P. (1999). *Operations management: strategy and analysis.* 5th ed. Addison-Wesley, Reading, Massachusetts, USA.

Lane, D.C. (2000). You just don't understand me: modes of failure and success in the discourse between system dynamics and discrete event simulation. *LSE OR WORKING PAPER 00.34,* London, UK.

Lanner. (2003). *Witness* [CD-ROM], ver. 2002, [Computer programme]. Available distributor: Lanner Group Limited, The Oaks, Clews road, Redditch, Worcestershire, B98 7ST, UK.

Law, A.M. and Kelton, W.D. (1991). *Simulation modelling and analysis,* 2nd ed. McGraw Hill, New York, USA.

Lee, Y.H., Cho, M.K., Kim, S.J. and Kim, Y.B. (2002). Supply chain simulation with discrete-continuous combined modelling. *Computers & industrial engineering,* 43, 375-392.

Lehmer, D.H. (1951). *Proceedings of the Second Symposium on Large-Scale Digital Computing Machinery,* Harvard University Press, Cambridge, Massachusetts, USA. In: Banks, J., Carson, J.S., and Nelson, B.L. (1996). *Discrete-Event system simulation.* 2nd ed, Prentice Hall, New Jersey, USA.

Lin, C., Baines, T.S., O'Kane, J. and Link, D. (1997). A generic methodology that aids the application of System Dynamics to manufacturing system modelling. Lecture notes, Staffordshire University, UK.

Lopez, M. (2000). *An evaluation theory perspective of the architecture trade-off analysis method (ATAMsm).* CMU/SEI-2000-TR-012 technical report. Carnegie Mellon software engineering institute, Pittsburgh, USA.

Macias-Chapula, C.A. (1995). Development of a soft systems model to identify information values, impact and barriers in a health care information system. *Journal of Information Science,* 21, 283-288.

MacKenzie, G.R., Gordon, G., and Yilmaz, L. (2002). Verification technology potential with different modelling and simulation development and implementation paradigms. In: *Proceedings of the Foundations for V&V in the 21st century workshop, Foundations'02.* Laurel, Maryland, USA, 22-24 October 2002.

Manstead, A.S.R. and Semin, G.R. (1998). Methodology in social psychology: turning ideas into action. In: *Hewstone, M., Stoebe, W., Codol, J.P. and Stephenson, G.M. (eds), Introduction to social psychology.* Blackwell, Oxford, UK.

Martin, L.A. (1997a). The first step. *MIT working papers.* D-4694. Massachusetts, USA.

Martin, L.A. (1997b). An introduction to feedback. *MIT working papers.* D-4691. Massachusetts, USA.

Martin, L.A. (1997c). Beginner modelling exercises. *MIT working papers.* D-4347. Massachusetts, USA.

Martin, R. and Raffo, D. (2001). Application of a hybrid process simulation model to a software development project. *The Journal of Systems and Software*, 59, 237-246.

Mathworks. (2003). *MatLab* [CD-ROM], ver. 5.5, [Computer programme]. Available distributor: The MathWorks, Inc. 3 Apple hill drive, Natick, MA, USA.

Mason, D. and Willcocks, L. (1994). *Systems analysis, systems design.* Waller Publishing, UK.

Mason-Jones, R., Berry, D., and Naim, M.M. (1998). A systems engineering approach to manufacturing systems analysis. *Integrated manufacturing systems,* 9(6), 350-365.

Mayhew, D.J. (1992). *Principles and guidelines in software user interface design.* Prentice Hall, Englewood Cliffs, New Jersey, USA.

McConnell, S.C. (1996). *Rapid development: taming wild software schedules.* Microsoft Press, Seattle, USA.

McConnell, S.C. (1997). *Software project survival guide.* Microsoft Press, Seattle, USA.

McHaney, R. (1991). *Computer simulation: a practical perspective.* Academic Press, San Diego, USA.

McKay, K.N. (2003). Historical survey of manufacturing control practices from a production research perspective. *International Journal of Production Research.* 41(3), 411-426.

McLean, C., and Leong, S. (2001). The expanding role of simulation in future manufacturing. In: *Proceedings of the 2001 Winter Simulation Conference,* USA, December 8-11 2000, 1478-1486. ACM.

Meadows, D.H., Meadows, D.L., and Randers, J. (1972). *The limits to growth.* Universe books, New York, USA.

Meek, B.L., Heath, P.M., and Rushby, N.J. (1983). *Guide to good programming practice,* 2nd ed. Ellis Horwood, Chichester, UK.

Mehta, A. (2000). Smart modelling, basic methodology and advanced tools. In: *Proceedings of the 2000 Winter Simulation Conference,* Orland, USA, December 10-13 2000, 241-245. ACM.

Merriam-Webster's Unabridged Dictionary. (2000). Merrian-Webster, USA.

Mertins, K. Sussength, W. et al. (1992). An object oriented method for Integrated Enterprise Modelling as a basis for enterprise coordination. In: *Proceedings of the First International Conference of Enterprise Integration Modelling.* The MIT press, Austin, Texas, USA.

Meyers, F.E., and Stephens, M.P. (2000). *Manufacturing facilities design and material handling,* 2nd ed. Prentice-Hall, New Jersey, USA.

Microsoft Corporation. (2003a). *Visual Basic* [CD-ROM], ver. NET, [Computer programme]. Available distributor: MSN, Microsoft house, 10 great pulteney street, London W1R 3DG, UK.

Microsoft Corporation. (2003b). *C++* [CD-ROM], ver. NET, [Computer programme]. Available distributor: MSN, Microsoft house, 10 great pulteney street, London W1R 3DG, UK.

Microsoft Corporation. (2003c). *Assembler* [CD-ROM], ver. 7, [Computer programme]. Available distributor: MSN, Microsoft house, 10 great pulteney street, London W1R 3DG, UK.

Microsoft Corporation. (2003d). *Office* [CD-ROM], ver. XP, [Computer programme]. Available distributor: MSN, Microsoft house, 10 great pulteney street, London W1R 3DG, UK.

Microsoft Corporation. (2003e). *Fortran* [CD-ROM], ver. NET, [Computer programme]. Available distributor: MSN, Microsoft house, 10 great pulteney street, London W1R 3DG, UK.

Mihram, G.A. (1972). The modelling process. IEEE Transactions on Systems, Man and Cybernetics, smc-2(5), November 1972, 621-629.

Miles, M.B. and Huberman, A.M. (1984). *Qualitative data analysis: a sourcebook of new methods,* SAGE, Newbury Park, London, UK.

Mills, R.I. and Talavage, J.J. (1985). Simulation programs for FMS design. In: *Proceedings of SIM-1 Conference,* March 1985, Stratford-upon-Avon.

Mills, S. (1993). Simulation saves time and money. *Engineers Australia,* March 1993, 27-28.

Mintzberg, H., Raisinghani, D., and Theoret, A. (1976). The structure of "unstructured" decision process. *Administrative science quarterly*, vol. 21, pp. 246-275.

Mohapatra, P.K.J. (1980). Part1 – Structural equivalence between control system theory and system dynamics; Part II – Non-linearity in system dynamics models. *Dynamica*, 6(1).

Moore, R. (1999). *Making common sense common practice: models for manufacturing excellence.* Gulf, Houston, Texas, USA.

Morecroft, J.D.W. (1982). A critical review of diagramming tools for conceptualising feedback systems models. *Dynamica,* 8(1), 20-29.

Morris, G.E. (1977). *Engineering, a decision-making process.* Houghton Mifflin Company, Boston, USA.

Mujtaba, M.S. (1994). Enterprise modelling and simulation: complex dynamic behaviour of a simple model of manufacturing. *Hewlett Packard Journal,* October 1994, 80-112.

Mullet, K., and Sano, D. (1995). *Designing visual interfaces,* SunSoft Press, Prentice Hall, Englewood Cliffs, New Jersey, USA.

Myers, G.J. (1975). *Reliable software through composite design,* Van Nostrand Reinhold, Wokingham. In: Easteal, C., and Davies, G. (1989). *Software engineering: analysis and design.* McGraw-Hill, London, UK.

Nance, R.E. (1995). Simulation programming languages: an abridged history. In: *Proceedings of the 1995 Winter Simulation Conference,* Orland, USA, December 10-13 1995, 1307-1313. ACM.

Nance, R.E., and Sargent, R.G. (2002). Perspectives on the evolution of simulation. *Operations Research.* 50(1), 161-172.

Naylor, T.J., Balintfy, J.L., Burdick, D.S., and Chu, K. (1966). *Computer simulation techniques.* John Wiley and Sons, New York, USA.

Nellis, J., and Figueira, C. (2003). Is there a future for UK manufacturing?. *Management Focus,* 2003 issue, Cranfield School of Management, Cranfield, UK.

Nikolaidou, M., and Anagnostopoulos, D. (2003). A distributed system simulation modelling approach. *Simulation modelling practice and theory,* 11, 251-267.

Nikoukaran, J., Hlupic, V., and Paul, R.J. (1998). Criteria for simulation software evaluation. In: *Proceedings of the 1988 Winter Simulation Conference.* Washington DC, USA, 13-16 December 1998, ACM.

Nunnally, J.C. (1967). Psychometric theory, McGraw-Hill, New York, USA.

Oakshott, L. (1997). *Business modelling and simulation.* Pitman, London, UK.

O'Kane, J.F., Spenceley, J.R., and Taylor, R. (2000). Simulation as an essential tool for advanced manufacturing technology problems. *Journal of Materials Processing Technology,* 107, 412-424.

Okudan, G.E. and Kabadayi, E. (2001). Manufacturing performance management in a competitive environment: a survey study at Gebze industrial park. In: *Proceedings of the 29th Computers and Industrial Engineering conference (ICC&IE).* Montreal, Canada, Nov 1-3 2001, 391-399.

Olinick, M. (1978). *An introduction to mathematical models in the social and life sciences.* Addison-Wesley Publishing Company, Massachusetts, USA.

Olsmats, C.M.G., Edghill, J.S., and Towill, D.R. (1988). Industrial dynamics model building of a close-coupled production distribution system. *Engineering costs and production economics,* 13, 295-310.

Online Learning Resource. (2003). *Importance of manufacturing industry* (WWW document). http://www.tutor2u.net/economics/content/topics/manufacturing/manufacturing_importance.htm. (Accessed 19th May 2003).

Oppenheim, A.N. (1992). *Questionnaire design, interviewing and attitude measurement,* 2nd ed. Printer publishers, London, UK.

O'Reilly, S.B., Casey, K.W., and Weiner, S.A. (1984). On the use of physical models to simulate assembly plant operations. In: *Proceedings of the 1984 Winter Simulation Conference,* USA.

Ossimitz, G. (2001). *Teaching system dynamics and systems thinking.* (WWW document). www.uni-klu.ac.at/users/gossimit/eng/sdyneng.htm. (Accessed 19th May 2002).

O'Sullivan, D. (1991). Project management in manufacturing using IDEFo. *International Journal of Project Management,* 9(3), 162-168.

Oyarbide-Zubillaga, A. (1999). *Investigation into the application of System Dynamics into manufacturing systems design.* MSc thesis. Cranfield University, Cranfield, UK.

Oyarbide-Zubillaga, A., Baines, T.S., Kay, J.M., and Ladbrook, J. (2000a). 'A system dynamics simulator for manufacture'. In: *Proceedings of the 1st World conference on production and operations management*, University of Seville, Seville, Spain, August 27th - September 1st 2000, part L1S01, 7p.

Oyarbide-Zubillaga, A., Baines, T. S., Kay, J. M., and Ladbrook, J. (2000b). Development of a manufacturing simulator based on system dynamics principles. In: *Proceedings of the 33rd International Symposium on Automotive Technology and Automation: ISATA 2000*, Dublin, Ireland, September 25-29 2000, 95-101. Epsom, UK.

Oyarbide-Zubillaga, A., Baines, T.S., Kay, J.M. and Ladbrook, J. (2003). Manufacturing systems modelling using system dynamics: forming a dedicated modelling tool. *Journal of Advanced Manufacturing Systems*, 2(1), June 2003, 71-87.

Page, E.H. and Nance, R.E. (1994). Parallel discrete event simulation: a modelling methodological perspective. In: *Proceedings of the ACM/IEEE/SCS 8th Workshop on Parallel and Distributed Simulation.* July 6-8 1994, 88-93, Edinburgh, Scotland.

Parasuraman, A. (1986). *Marketing research.* Addison Wesley. Reading, Massachusetts, USA.

Paxson, M.C. (1992). Follow-up mail surveys. *Industrial marketing management,* 21, 195-201.

P-E International. (2003). *HOCUS Simulation 8000* [CD-ROM], ver. 2003, [Computer programme]. Available distributor: P-E International, P.O. Box 840 Purcellville, VA, USA.

Perera, T., and Liyanage, K. (2000). Methodology for rapid identification and collection of input data in the simulation of manufacturing systems. *Simulation practice and theory.* 7, 645-656.

Petropoulakis, L., and Giacomini, L. (1998). Development of a hybrid simulator for manufacturing processes. *Computers in industry,* 36, 117-124.

Petroutsos, E. (2000). *Mastering database programming with Visual Basic 6.* Sybex, San Francisco, USA.

Petroutsos, E., and Hough, K. (1999). *Visual Basic 6, developer's handbook.* Sybex, San Francisco, USA.

Phillips, E.M. and Pugh, D.S. (2000). *How to get a PhD: a handbook for students and their supervisors,* Open University Press, London, UK.

Pidd, M. (1988). *Computer simulation in management science,* 4th ed. John Wiley and Sons, Chichester, UK.

Pidd, M. (1999). Just modelling through: a rough guide to modelling. *Interfaces,* 28(2), 118-132.

Potter, J. (2000). *On the effectiveness and efficiency of Discrete Event Simulation for designing manufacturing systems.* EngD thesis, Cranfield University, Cranfield, UK.

Powell, S.G., Schwaninger, M., and Trimble, C. (2001). Measurement and control of business processes. *System Dynamics review,* 17(1), 63-81.

Powersim Software. (2003). *Powersim Studio* [CD-ROM], ver. 2003, [Computer programme]. Available distributor: Powersim Software, PO Box 3961 Dreggen, N-5835 Bergen, Norway.

ProModel Solutions. (2003). *ProModel* [CD-ROM], ver. 5.4, [Computer programme]. Available distributor: ProModel Solutions, 1875 S. State Street, Suite 3400 Orem, UT 84097, USA.

Pugh-Roberts Associates. (2003). *Dynamo* [CD-ROM], ver. Professional Plus, [Computer programme]. Available distributor: Pugh-Roberts Associates, 41 William Linskey Way, Cambridge MA 02142, USA.

Rabbath, C.A., Abdoune, M., and Belanger, J. (2000). Effective real-time simulations of event-based systems. In: *Proceedings of the 2000 Winter Simulation Conference,* Orland, USA, December 10-13 2000, 232-238. ACM.

Radzicki, M.J. (1997). Foundations of system dynamics modelling. *Sustainable solutions.*

Rathwell, G. (2000). *IDEF structured systems analysis diagrams.* (WWW document). http://www.pera.net/Standards/Stds_IDEF.html. (Accessed 19th May 2003).

Rauniar, R., Rawski, G., and Shah, S. (2002). Simulation software selection for manufacturing: a review. In: *Proceedings of the Decision Science Instutitute 2002 Annual Meeting,* Toledo, Ohio, 2013-2016, University of Toledo.

Richardson, G. and Pugh, A.L. (1981). *Introduction to System Dynamics Modelling with DYNAMO.* MIT Press, Cambridge, Massachusetts, USA.

Richmond, B. (1994). Systems thinking / system dynamics: let's get on with it. *System Dynamics Review*, 10(2-3), 135-157.

Roberts, E.B. (1978). Strategies for effective implementation of complex corporate models. In: Roberts, E.B. (1978). *Managerial applications of System Dynamics*. 77-85. MIT Press, Cambridge, Massachusetts, USA.

Roberts, N., Andersen, D., Deal, R., Garet, M., and Shaffer, W. (1983). *Introduction to computer simulation: a System Dynamics modelling approach.* Addison-Wesley, Massachusetts, USA.

Robinson, S. (1994). *Successful simulation: a practical approach to simulation projects.* McGraw Hill, London, UK.

Robson, C. (1993). *Real world research: a resource for social scientists and practitioners,* Blackwell, Massachusetts, USA.

Robson, C. (2002). *Real world research: a resource for social scientists and practitioner-researchers,* 2nd ed, Blackwell, Massachusetts, USA.

Rockwell Software. (2003). *Arena* [CD-ROM], ver. 7.01, [Computer programme]. Available distributor: Rockwell Automation, US Bank Center, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, Wisconsin 53202, USA.

Roth, P.F. (1987). Discrete, continuous and combined simulation. In: *Proceedings of the 1987 Winter Simulation Conference,* 25-29, USA.

Roy, R. (1997). *Adaptive search and the preliminary design of gas turbine blade cooling system.* PhD thesis. University of Plymouth, Plymouth, UK.

Royce, W. (1998). *Software project management: a unified framework.* Addison Wesley, Massachusetts, USA.

Rubinstein, R.Y., and Melamed, B. (1998). *Modern simulation and modelling.* John Wiley and Sons, New York, USA.

Sanjek, R. (2000). Keeping ethnography alive in an urbanised world. *Human organisations.* Fall.

Sarjoughian, H.S., and Zeigler, B.P. (2000). Models and representation of their ownership. In: *Proceedings of the 2000 Winter Simulation Conference,* Orland, USA, December 10-13 2000, 440-448. ACM.

Sastry, M.A. (1993). The Dynamics of Organizational Change: A Simulation-Based Exploration of a Punctuated Equilibrium Theory. MIT D-4318 working paper.

Saunders, M., Lewis, P. and Thornhill, A. (1997). *Research methods for business students,* Pitman Publishing, London, UK. In: Sherwin, C. (2000). *Innovative ecodesign: an exploratory and descriptive study of industrial design practice.* PhD thesis. Cranfield University, Cranfield, UK.

Sawhney, A., Manickam, J., Mund, A., and Marble, J. (1999). Java-based simulation of construction processes using Silk. In: *Proceedings of the 1999 Winter Simulation Conference.* Phoenix, USA, December 1999, 985-991, ACM.

Schaefer, L.A., and Wolfe, P.M.. (2000). Issues in Java-based continuous time step physical modelling. In: *Proceedings of the 2000 Winter Simulation Conference,* Orland, USA, December 10-13 2000, 1882-1887. ACM.

Scheele, N. (2002). The importance of UK manufacturing. *We can make it better, manufacturing 2020 consultations recommendations report.* Foresight, UK.

Schmidt, J.W. (1985). Introduction to systems analysis, modelling and simulation. In: *Proceedings of the 1985 Winter Simulation Conference.* San Francisco, California, USA, 3-14. ACM.

Schrage, M. (1999). *Serious play: how the world's best companies simulate to innovate.* Harvard Business School Press, Cambridge, Massachusetts, USA.

Seveance, F.L. (2001). *Systems modelling and simulation: an introduction.* John Wiley and Sons, New York, USA.

Shannon, R.E. (1975). *Systems simulation, the art and science.* Prentice Hall, London, UK.

Shannon, R.E. (1998). Introduction to the art and science of simulation. In: *Proceedings of the 1998 Winter Simulation Conference*, Washington DC, USA, December 13-16, 1998, 7-14p. ACM.

Shreckengost, R.C. (1985). Dynamic simulation models. How valid are they? In: Rouse, B.A., Kozel, N., and Richards, L.G. (eds). *Self reporting methods of estimating drug use: meeting current validity challenges,* NIDA Research monograph 57, 63-70.

Siebers P.O. (2001). Integrated human performance modelling; a overview of modelling and its usability in manufacturing system design. In: *Proceedings of the Operational Research Society, Annual Conference*, 4-6 September, University of Bath, UK..

Simulation Study Group. (1991). *Simulation in the UK manufacturing industry.* DTI study report, Coventry, UK.

Sivayoganathan, K., Al-Dabass, D., and Balendran, V. (2001). Simulation: what is missing? In: *Proceedings of the United Kingdom Simulation Society Conference UKSIM'01.* Cambridge, UK, 28-30 Mar 2001.

Slack, N., Chambers, S., Harland, C., Harrison, A., and Johnston, R. (1998). *Operations management,* 2nd ed. Prentice Hall, London, UK.

Smith, R.V. (1990). *Graduate research,* Plenum press, London, UK.

Sokolow, H. (1985). In-depth interviews increasing in importance. *Marketing news,* September 13th, 26.

Spur, G., Mertins, K., and Jockem, R. (1996). *Integrated enterprise modelling,* Beuth Verlag GmbH, Berlin, Germany.

Stanley, L. and Zhu, H. (1996). Mental simulation of combining feedback in first-order systems. *MIT working papers.* D-4593. Massachusetts, USA.

Sterman, J.D. (1985). A sceptic's guide to computer models. In: *Proceedings of the International System Dynamics Conference,* July 1985, 45p.

Sterman, J.D. (1991). A sceptic's guide to computer models. In: Barney, G.O. *et al.* (eds.). *Managing a nation: the microcomputer software catalogue,* 209-229. Westview Press. Boulder, CO.

Sterman, J.D. (2000). *Business dynamics: systems thinking and modelling for a complex world.* Irwin/McGraw-Hill, Boston, USA.

Suh, N.P. (1990). *The principles of design.* Oxford University Press, USA.

Sule, D.R. (1994). *Manufacturing facilities: location, planning and design.* PWS, USA.

Sun Microsystems, Inc. (2003). *Java Enterprise* [CD-ROM], ver. 2, [Computer programme]. Available distributor: 4150, Network Circle, Santa Clara, CA 95054, USA.

Suri, R., and Diehl, G.W. (1985). A precursor to simulation for complex manufacturing systems. In: *Proceedings of the 1985 Winter Simulation Conference,* 441-420. ACM.

Tarr, J.D. (2001). *Activity Based Costing in the information age* (WWW document). http://www.theacagroup.com/activitybasedcosting.htm. (Accessed 19th May 2003).

Tempelmeier, H., and Kuhn, H. (1993). *Flexible manufacturing systems: decision support for design and operations.* John Wiley and Sons, Chichester, UK.

Terzis, S. (2001). *Systems analysis and design. University of Strathclyde* (WWW document). http://www.cs.strath.ac.uk/~terzis/classes/52.234/2001. (Accessed 19th May 2003)

Tiwari, A. (2002). *Evolutionary computing techniques for handling variable interaction in engineering design optimisation.* PhD thesis, Cranfield University, Cranfield, UK.

Tocher, K.D. and Owen, D.G. (1960). The automatic programming of simulations. In: *Proceedings of the Second International Conference on Operational Research.* 50-68.

Tocher, K.D. (1979). Keynote address. In: *Proceedings of the 1979 Winter Simulation Conference,* December, 3-5, 640-654, San Diego, California, USA.

Tognazzini, B. (2000). *If they don't test, don't hire them* (WWW document). http://www.asktog.com/columns/037TestOrElse.html. (Accessed 19th May 2003).

Trybula, W.J. (1994). Building simulation models without data. In: *Proceedings of the IEEE International Conference of Systems, Man and Cybernetics*, 209-201. Cited in: Liyanage, K. and Perera, T. (1997). Rapid data modelling techniques in simulation of manufacturing systems. In: *Proceedings of the 5th International Conference on FACTORY 2000,* April, 2-4, 178-182.

Tull, D.S. and Hawkins, D.I. (1990). *Marketing research: measurement and method.* Macmillan, New York, USA.

Turney, P.B. (1996). *Activity Based Costing: the performance breakthrough.* Kogan Page Limited, London, UK.

Van Beek, D.A., and Rooda, J.E. (2000). Languages and applications in hybrid modelling and simulation: positioning of Chi. *Control Engineering Practice,* 8(1), 81-91.

Vemuri, V. (1978). *Modelling of complex systems, an introduction.* Academic Press, New York, USA.

Ventana Systems. (2003). *Vensim* [CD-ROM], ver. 5.1b, [Computer programme]. Available distributor: Ventana Systems, Inc., 60 Jacob Gates Road, Harvard, MA 01451, USA.

Voss, C., Tsikriktsis, N. and Frolich, M. (2002). Case research in operations management. *International Journal of Operations and Production Management,* 22(2), 195-219.

Wang, W., and Skeel, R.D. (2003). Analysis of a few numerical integration methods for the Langevin equation. *Molecular physics,* 101(14), 2149-2156.

Wideman, R.M. (2003). Software development and linearity. *Journal of Projects and Profits.* March 2003 (Special issue), 18-33.

Williams, T. (2000). *Enterprise models and methodologies* (WWW document). http://www.pera.net/Ind_meth.html. (Accessed 19th May 2003).

Wolstenholme, E.F. (1990). *System enquiry: a System Dynamics approach.* John Wiley and Sons, Chichester, UK.

Wu, B. (1994). *Manufacturing systems design and analysis: context and techniques.* Kluwer Academic Publishers, New York, USA.

Wu, B. (2000). *Manufacturing and supply systems management: a unified framework of systems design and operation.* Springer-Verlag, UK.

Xu, J., and Abourizk, S. (1999). Product-based model representation for integrating 3D CAD with computer simulation. In: *Proceedings of the 1999 Winter Simulation Conference.* Phoenix, USA, December 1999, 971-977, ACM.

Yin, R.K. (1994). *Case study research: design and methods,* SAGE Publications, Thousand Oaks, California, USA.

Young, R.E., Campbell, J.A., and Morgan, J.A. (1984). Physical simulation: the use of scaled-down fully functional components to analyze and design automated production systems. *Computers and Industrial Engineering,* 8(1), 73-85.

Yourdon, E. and Constantine, L.L. (1979). *Structured design: fundamentals of a discipline of computer program and system design,* Prentice-Hall, Englewood Cliffs, New Jersey, USA. In: Easteal, C., and Davies, G. (1989). *Software engineering: analysis and design.* McGraw-Hill, London, UK.

Zaraza, R. (1998). The CC-STADUS 12-Day Training Materials. In: *A comprehensive workshop designed to prepare teachers to develop computer models and curriculum for use in their classrooms.* CC-STADUS/CC-Sustain Institute, summer 1998, Portland.

Zelkowitz, M.V., Shaw, A.C., and Gannon, J.D. (1979). *Principles of software engineering and design.* Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Zhu, H. (1996). Mental simulation of simple negative feedback. *MIT working papers.* D-4536. Massachusetts, USA.

Zobel, R.N. (2001). Advances in distributed simulation. In: *Proceedings of the United Kingdom Simulation Society Conference UKSIM'01.* Cambridge, UK, 28-30 Mar 2001.

Zopounidis, C. (2002). MCDA methodologies for classification and sorting. *European Journal of Operation Research,* 138(2), 227-228.

Appendix A

SYSTEMS PRINCIPLES

The following list exhibits the 'system principles' developed by Forrester:

The feedback loop is the basic structural elements of systems.

- Simplex systems are composed of positive and negative feedback loops, although more complex feedback loops can exist in a system. These feedback loops are used as the building blocks and are linked together to build more complex systems.

Levels and Rates are fundamental to loop substructure.

- A feedback loop consists of two distinct types of variables, the levels (also called stocks or states) and the rates (also called flows or actions). These two variables are both necessary and sufficient to represent the structure in a feedback loop.

Levels and Rates are not distinguished by units of measure.

- Units do not determine whether a variable is a level or a rate. For example, 'velocity' can either be a rate that increases the distance or can be a level that accumulates acceleration. In both cases, 'velocity' is in units of distance per time.

Levels are accumulations (integration).

- Levels accumulate the results of rates (actions) in the system. Levels change smoothly but not instantaneously (there are no discontinuities or jumps).

Levels are changed only by the Rates.

- Rates change variables. But no other variables alter the levels. Even the levels do not alter each other or themselves. A level variable's current value is computed using only its previous value and the change due to the rates acting on the level. The earlier value of the level is carried forward from the previous period. It is altered by rates that flow in and out of the level over the intervening time period. The present value of a level is not directly dependent on the present or previous values of any other levels.

Levels exist in conservative subsystems.

- A conserved quantity has the property that it is never created or destroyed (within its system); it is only moved around.

Rates depend only on Levels and Constants.

- Two rates cannot directly influence each other. The value of a rate variable depends only on present values of level variables and constants. No rate variable depends directly on any other rate variable. The rate equations of a system are of simple algebraic form; they do not involve time or the solution interval; they are not dependent on their own past values.

Decisions are always within feedback loops.

- Every decision process is made within at least one feedback loop.

Every equation must have dimensional equality.

- In any equation, every term must be measured in the same dimensions. Dimensional inequality between terms indicates a faulty equation formulation.

First-order loops exhibit exponential behaviour.

- The first-order feedback loop always exhibits an exponential time shape.

Levels completely describe the system condition.

- The values of all other variables (the rate variables) can be computed from these values and the system equations alone.

Variables have the same unit within conservative subsystems.

- Recall from 'System principle 6' that levels exist in conservative systems, the contents of stocks are neither created nor destroyed, just transferred between levels via flows. Levels connected within a conservative subsystem have the same units of measurement.

Solution interval DT is in all level equations and no others.

- The DT (also called the solution interval, period of measurement, delta time, or time step) is the time period in which the level is changed by the rate. The DT is essential to the level equation.

Simple, second-order negative loops exhibit sinusoidal oscillation.

- The oscillation is independent of the values of parameters; this is because it has the same qualitative structure. Any second-order negative loop with no minor loops oscillates as a sustained sinusoid.

Goal, observation, discrepancy and action create a system substructure.

- A policy or rate equation recognises a local goal toward which the decision point strives, compares the goal with the apparent system condition to detect a discrepancy, and uses the discrepancy to guide action.

Level variables and Rate variables must alternate.

- Any part through the structure of a system encounters alternating level and rate variables. For any loop in a system, if it starts at a level variable, the next variable cannot be another level; the next variable must be a rate.

Higher-order, positive-feedback loops usually show exponential behaviour.

- ➢ Positive feedback loops of n^{th} order usually exhibit simple exponential growth (ignoring possible initial transients). In most real-world systems, the initial values are such that the positive feedback loops will generate exponential growth.

Conversion coefficients are identifiable within real systems.

- ➢ Conversion coefficients should always have a clear, real meaning. They are not inserted merely to balance equations. They should have numerical values that can be logically deduced from observation. They are not only the result of statistical analysis.

Time constant of a first-order loop relates a level to a rate.

- ➢ The exponential time constant of a first-order loop is reciprocal of the multiplier that defines the rate in terms of the level. It relates a level to the rate that affects it. The 'rate' is equal to the 'level' divided by the 'time constant', or to the 'level' multiplied by the reciprocal of the 'time constant'.

Rates are not instantaneously measurable.

- ➢ No rate of flow can be measured instantaneously. A rate is a change over time. Without an observation over a time interval, a rate cannot be measured.

Every system has a closed boundary.

- ➢ In creating a model of a real system, any interaction that is essential to the behaviour mode being investigated must be included inside the system boundary. If a model is to generate the same behaviour as the real system, then the system structure that is responsible for that behaviour must be included inside the model. The behaviour and its generator are endogenous to the closed system.

Information links connect levels to rates.

- Information links, or connectors, link levels to the control of rates. Through information links, values of level variables go to the rate equations, determining the rates of flow.

Decisions (rates) are based only on available information.

- Decisions are made based on the policy statements in the rate equations. The rate equations in a System Dynamics model are policy statements that determine how 'decisions' are made.

Auxiliary variables lie only in the information links.

- An auxiliary variable, or converter, is a subdivision of a rate equation. It allows a model to be desegregated into easier to understand equation statements.

Mathematical simulation models belong to the broad class of abstract models.

- A model is a substitute for an object or a system. Some models are physical, such as a toy aeroplane or an architectural scale model. We are familiar with these. Some models are abstract. These abstract models include mental images, literary descriptions, behaviour rules for games, and legal codes. Mathematical simulation models also belong to the broad class of abstract models. Because computer modelling has become so widespread in recent years, it is important to understand the assumptions and applications of various modelling techniques.

Model validity is a relative matter.

- The usefulness of a mathematical simulation model should be judged in comparison with the mental image or other abstract model that would be used instead. No model is a perfect representation of a real object. A model is successful if it opens the road to improving the accuracy with which the reality can be represented.

Appendix B

STAGES OF A TYPICAL SIMULATION PROJECT

The main stages involved in a simulation project are briefly defined below. It is not intended to be a comprehensive discussion, but merely a general guide in order to obtain information that will be used for the design and development of the tool.

Stage 1 - Problem formulation, objectives and plan

If a problem is not formulated correctly, the solution will always be wrong (Guasch and Piera, 2001). In addition, as stated by Shannon (1975) *"millions of dollars are spent each year in coming up with elegant and sophisticated answers to the wrong questions"*. Thus, communication between the simulation analyst and the customer is essential in this stage. In addition, clear, unambiguous and feasible objectives will help to specify the boundaries and determine the assumptions of the project, and therefore build a model that is designed to solving that specific problem. Care must be taken not to make an erroneous assumption when defining the problem.

Stage 2 - Model conceptualisation

This stage reduces the real system to a logical flow diagram (Shannon, 1975). Simply stated, a simulation model captures the time it takes to do things. Thus, this stage involves the specification of the model by considering not only the system's characteristics and its interactions, but also the problem objectives in order to minimise the complexity of the model. Typical modelling elements defined in this stage include: resources, flow items (products, customers or information), routings, item transformations, flow control, process times, and resource down times. Translating reality into a model always means that you are providing an interpretation of reality. Thus, it is usually necessary to specify the assumptions that are made in the translation. However, the greater the effort invested in this stage, the simpler will be the step of building the computer model.

Stage 3 - Data collection

Simulation models require input data to create results. Data usually falls into one of three categories (Mehta, 2000): (i) Available, (ii) Not available, but collectable and (iii) Not available and not collectable. In practice, data are usually collected from historical records, experience or by calculation to use as input parameters to the model. However, it is also useful to collect real data that will allow the simulation analyst to validate the model and compare the performance measures of the model with the real ones. Existing sources of data are not always available, and data collection through measurements can be both expensive and time consuming. For example, if reliable data are needed, it is necessary to collect a statistically significant amount of data over a representative amount of time in order to define a probability distribution that accurately represents reality.

Stage 4 - Model translation / building

This stage is conducted using the previous stages as inputs. In practice, it is convenient to build the model modularly; running and debugging each sub-model individually, specially the complex ones. In very complex models, it is even advisable to increase the complexity of the model gradually, by first building a simple model, and adding complexity by stages.

Stage 5 - Model verification

Model verification is used simply to determine if the model functions as intended (Guash and Piera, 2001). There are a number of techniques that can be used to verify a simulation model. A typical one is to view the animation and simulation clock simultaneously while running the model in slow speed. This should point out any mayor discrepancies in flow routes and processing times. Another verification technique is to query the states and attributes of the resources and flow items in the model through the use of the interactive command window, or by displaying dynamic charts and graphs on the display screen while the model is running.

Stage 6 - Model validation

This stage is often more extensive and complex than the one for verification. It involves determining if the model is a correct representation of reality, and determining how much confidence can be placed in the results of the model (King, 2001). However, in practice, it cannot be proved that the behaviour of the model is an exact description of reality. In a strict sense, the subject of dynamic simulation model validity can be treated thoroughly and quickly: *"there are no fully valid models because all models are something less than the object, or system, being modelled"* (Shreckengost, 1985). Thus, a model is considered to be valid if it meets the objective by providing relatively accurate information. Coyle and Exelby (2000) explain the concept of validity and provide a review process for the validation of commercial SD models, while Sterman (1985) suggests a number of tests oriented to increase the confidence in simulation models. These are divided in three areas, and are explained below:

- **Tests of model structure:** Structure verification, Parameter verification, Extreme conditions, Boundary adequacy and Dimensional consistency.
- **Tests of model behaviour:** Behaviour reproduction, Behaviour anomaly, Family member, Surprise behaviour, Extreme policy, Boundary adequacy, Behaviour sensitivity and Statistical character.
- **Tests of policy implications:** System improvement, Behaviour prediction, Boundary adequacy, Policy sensitivity.

Stage 7 - Experimental design and runs

Experimental designs help to ensure that simulation runs are focussed on solving the problem stated in the first stage and also avoiding redundancies, in order to reduce the number of experiments that need to be conducted. Multiple simulation runs (or observations) are always required when stochastics are involved. When choosing the run length of the simulation, it is important to consider warm-up periods, and any other system characteristics that would require a long run length in order to capture the effect (breakdowns, seasonal variances, etc.). In most modern simulation tools, the alternative scenarios can be set up individually and simulated manually or automatic runs can be executed using optimization modules. To conduct an optimization, it is usually necessary to define an objective variable to be maximized or minimized, as many decision variables as desired, any requirements that need to be met, and any linear constraints that need to be satisfied.

Stage 8 - Analysis of experiments

This stage often aims to detect possible problems and suggest improvements or new solutions (Guash and Piera, 2001). When analyzing results and drawing conclusions, it is important to interpret the results in such a way that they relate to the objective. Reports, charts, graphs, and confidence interval plots often help the analysis. In addition, statistical techniques are also often used to analyze the output data from each of the alternative scenario runs. Because of the stochastic nature of typical real systems, a confidence interval is generally used, indicating the range in which the performance measure lies. The degree in which the upper and lower limits are separated is called the accuracy.

Stage 9 - Documentation and reporting

This stage provides the customer with detailed documentation about the project undertaken, including, for example, information about the problem definition and objectives, assumptions adopted, model developed, experimental design, analysis of results and recommendations. If the model is going to be used by the customer, it also usually also includes guidelines about its use (Shannon, 1975). In addition, documenting a simulation project is also useful to the simulation analyst, since it can reduce the time required to expand the model in future, or to solve similar problems.

Stage 10 - Implementation and/or training

The final stage of a simulation project concerns the customer, and consists of implementing (or not) the suggestions obtained from the simulation analyst and/or using the model for training purposes to increase the knowledge about the system.

Appendix C

INTERVIEW – PRESENTATION LETTER

{Date}
{Mr/Mrs/Ms}
{Job title}
{Company name}
{Address}

Simulation practices information survey

Dear {Name}

My name is Aitor Oyarbide and I am writing from Cranfield University. Your {Company/University} came to our attention through conversation with a colleague from {Mondragon University/Cranfield University/Ford Motor Company} working in the simulation area. Our work is broadly concerned with simulation and, in particular, competitive advantage gained through the application of a simpler simulation method based on the System Dynamics methodology. Hence, there appeared to be a strong resonance of our work with what {Company name} is doing in practice, and we are very interested in speaking to your company about these issues.

The current stage of the research requires us to conduct interviews with experienced {simulation practitioners/academic members in the area of simulation/simulation software developers}. I am extremely interested in {Company name} and felt it would add greatly to our research study. I would therefore like to ask if you would allow us to conduct an interview at {Company name}.

Let me explain what will be involved. First, I will provide you with a short presentation explaining the aim of the research and the objectives of the interview. Then, the interview will involve speaking to a member of your organisation, based on a structured, yet informal questionnaire. This interview is conducted with academic purposes and all information provided is confidential and no individual or organisation is identified.

For {Company name}, participation in the research will be an opportunity for learning with us; feedback will also be provided in the form of a written report, if required.

I hope you are interested in participating in this innovative research and I will call you shortly on the matter. In the meantime, if you have any questions, do not hesitate to telephone on: 01234 750111 ext. 2413 or email at a.oyarbide@cranfield.ac.uk.

Yours sincerely,

Aitor Oyarbide

School of Industrial and Manufacturing Science

Cranfield University, Cranfield, MK43 0AL, UK.

Appendix D

INTERVIEW QUESTIONNAIRE

This questionnaire aims to capture the key characteristics that industry requires when simulating their processes, in order to evaluate them and construct the requirements of the tool. It consists of five sections, which cover the following areas:

1. What do practitioners simulate / what should a SD tool should simulate?
2. How practitioners simulate / how should they simulate?
3. What results do they get from models?
4. What type of analysis do they do to the models?
5. What resources are involved in modelling a system?

The questions that form part of the questionnaire are given below. The explanations that are provided to the interviewees are also presented here. Answers are recorded by the interviewer in the boxes provided in this questionnaire and will be compiled and analysed using a computer tool.

Company:	
Interviewee:	
Date:	

Section 1: What practitioners simulate and what the tool should simulate.

This section evaluates which are 'elements' and 'scenarios' the interviewees consider most relevant. The proposed elements are taken after the analysis of several DES simulation packages.

Q 1a: Evaluating model's elements

How would you rate the importance of the following element when developing a model? (E=Essential; I=Important; A=Avoidable)

Element	Rate	Comments
Part		
Buffer		
Machine		
Labour		
Conveyor		
Shift		
Vehicle		
Track		
Av. Material flows		

Q 1b: Evaluating model's flows

How would you rate the importance of the following type of flows when developing a model? (E=Essential; I=Important; A=Avoidable)

Flow	Rate	Comments
1 to 1		
1 to N		
N to 1		
N to N		
Batch		

Q 1c: Evaluating model's advanced features

How would you rate the importance of the following features when developing a model? (E=Essential; I=Important; A=Avoidable)

Feature	Rate	Comments
Constant		
Variable		
Random number		
Breakdown pattern		
Setup pattern		

Q 1d: General comments about S1: 'What practitioners simulate'

Section 2: How practitioners simulate and how they should simulate.

This section evaluates where interviewees spend their time when simulating manufacturing processes. Elements included in this question are taken from the literature review.

Q 2a: Evaluating the assigned time for each modelling stage

How long do you spend (on average, approximately) for each of the following tasks when carrying out a simulation project? ('time' or '% of time/total')

Task	Time	Comments
Data collection		
Model coding		
Model verification		
Model execution		
Model validation		
Model experimentation		
Model expansion		
Analysis of results		

Q 2b: General comments about S2: 'How practitioners simulate'

Section 3: Which are the results that practitioners get from the models they simulate.

This section evaluates the main results that practitioners obtain from the models they simulate and the importance of them. It also aims to gather information on the 'quality' of the results they expect to obtain.

Q 3a: Evaluating the variety of results

Which of the following type of results do you usually obtain from the models you simulate and how would you evaluate them? (E=Essential; I=Important; A=Avoidable)

Result	Rate	Comments
Rate		
Utilisation		
Contribution		
Throughput		
Lead time		
Work in progress		
Waiting time		

Q 3b: Evaluating the quality of results

What are your expectations from the results you get? Is it only necessary that they are 'effective' at taking decisions or do they also need to be 'accurate'? (1=Never; 2=Sometimes; 3=Most of the times; 4=Always)

Quality	Rate	Comments
Effective		
Accurate		

Q 3c: General comments about S3: ‘Variety of results’

Section 4: How practitioners use their models.

This section evaluates how interviewees analyse their models while they are developing them and the way they present their results.

Q 4a: Analysing the computer models

How do you evaluate the following types of analysis regarding their relevancy for achieving the objectives of your simulation project? (E=Essential; I=Important; A=Avoidable)

Analysis	Rate	Comments
Step by step		
Track parts		
Animations in 'real time'		
Variable simulation speed		
Modification of the resolution of the system		
Statistics		
Reports		

Q 4b: General comments about S4: 'How practitioners analyse'

Section 5: Resources used by the practitioners.

This section evaluates the importance of human and technical resources used by the practitioners when simulating processes.

Q 5a: Evaluating the assigned resources

How would you describe the requirements needed when simulating? Please, provide both human and technical comments (e.g. skills, learning curve, computer graphical/process performance, etc.).

Software used:	

Appendix E

TOOL'S SCREENSHOTS

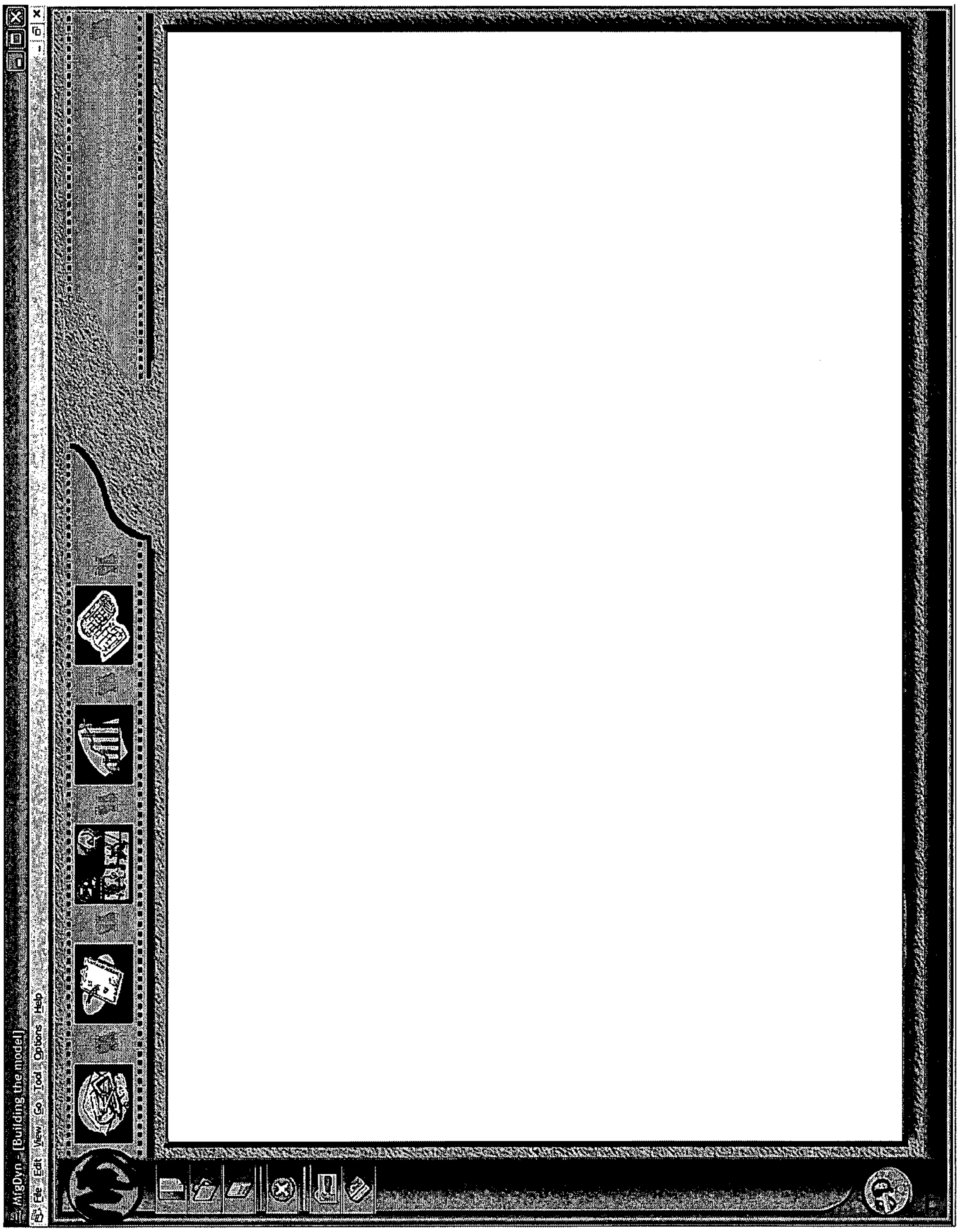


Figure 65: Screenshot of the container form

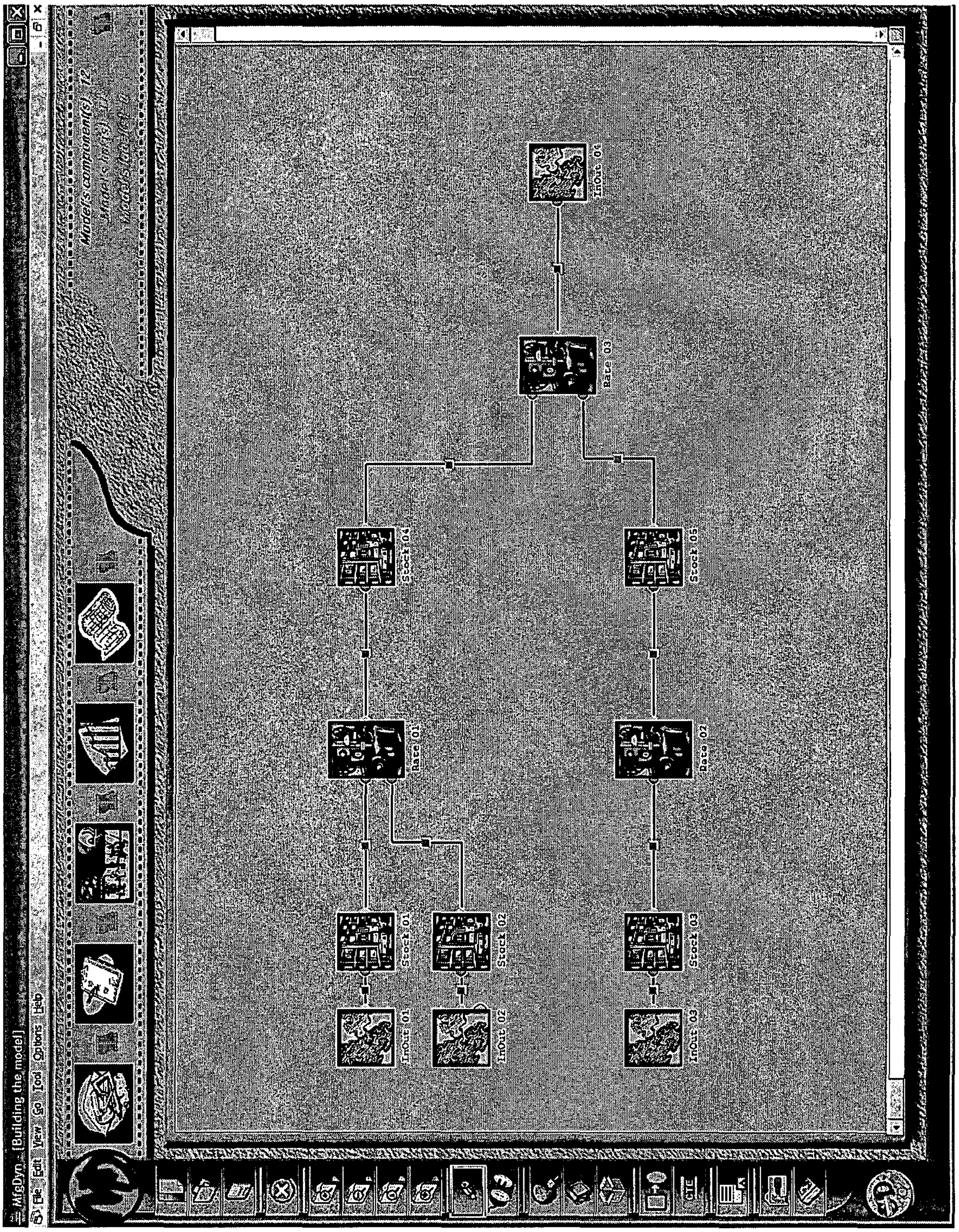
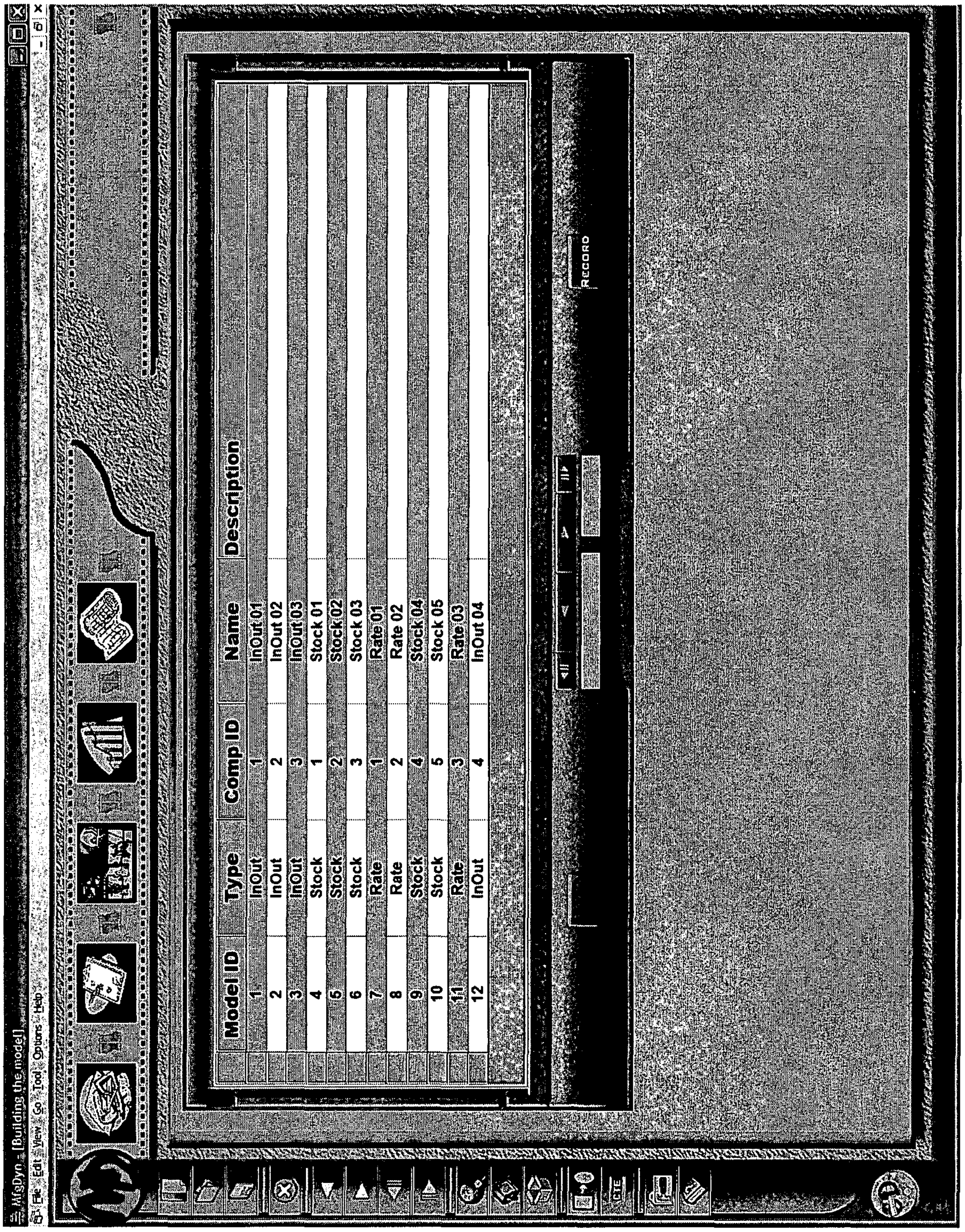
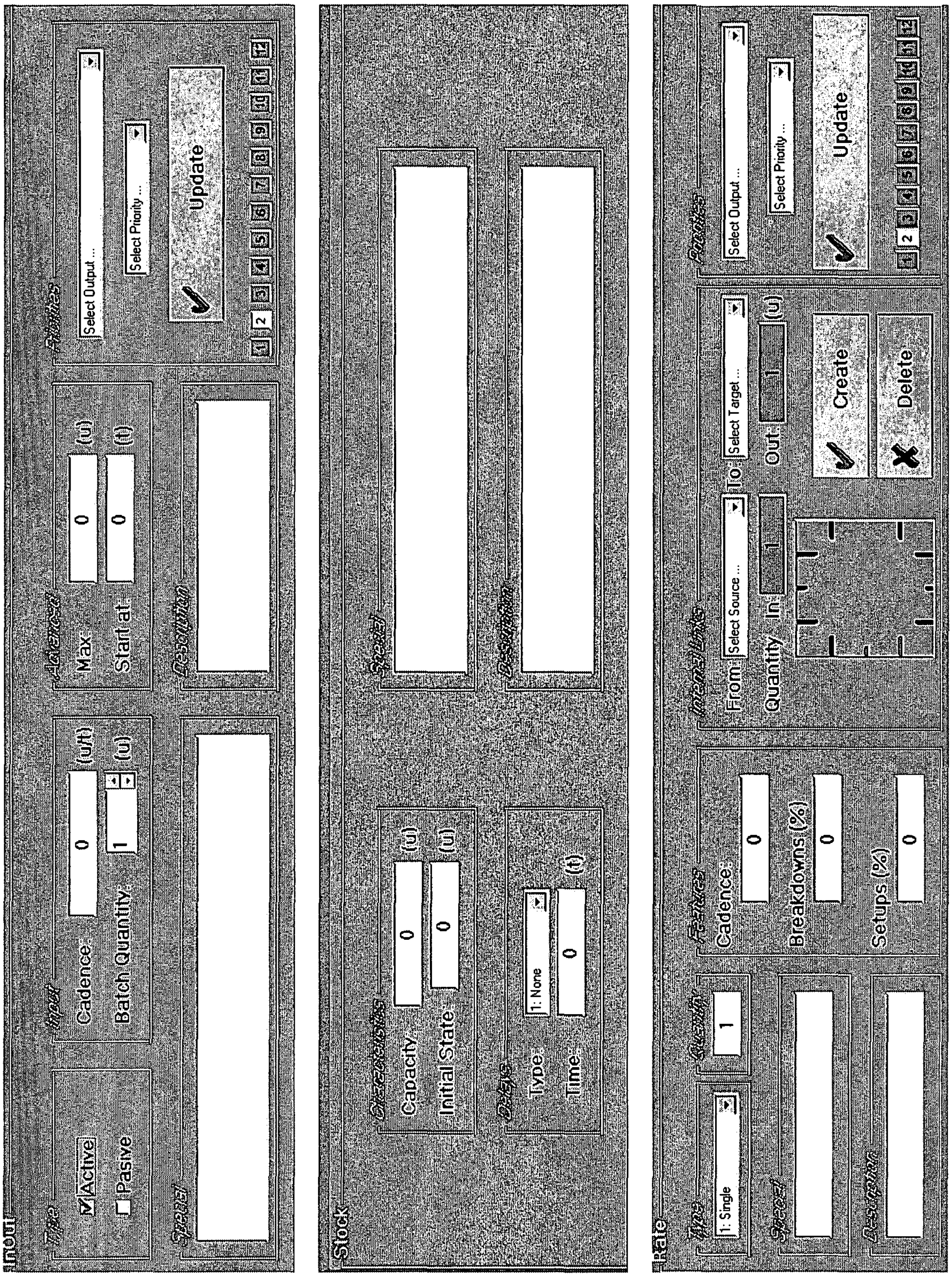


Figure 66: Screenshot of the model building form



Model ID	Type	Comp ID	Name	Description
1	InOut	1	InOut 01	
2	InOut	2	InOut 02	
3	InOut	3	InOut 03	
4	Stock	1	Stock 01	
5	Stock	2	Stock 02	
6	Stock	3	Stock 03	
7	Rate	1	Rate 01	
8	Rate	2	Rate 02	
9	Stock	4	Stock 04	
10	Stock	5	Stock 05	
11	Rate	3	Rate 03	
12	InOut	4	InOut 04	

Figure 67: Screenshot of the data introduction form



InOut

☑ Active
☐ Pasive

Cadence: 0 (u/t)
Batch Quantity: 1 (u)

Max: 0 (u)
Start at: 0 (t)

Select Output ...
Select Priority ...

Stock

Capacity: 0 (u)
Initial State: 0 (u)

Type: 1: None
Time: 0 (t)

Rate

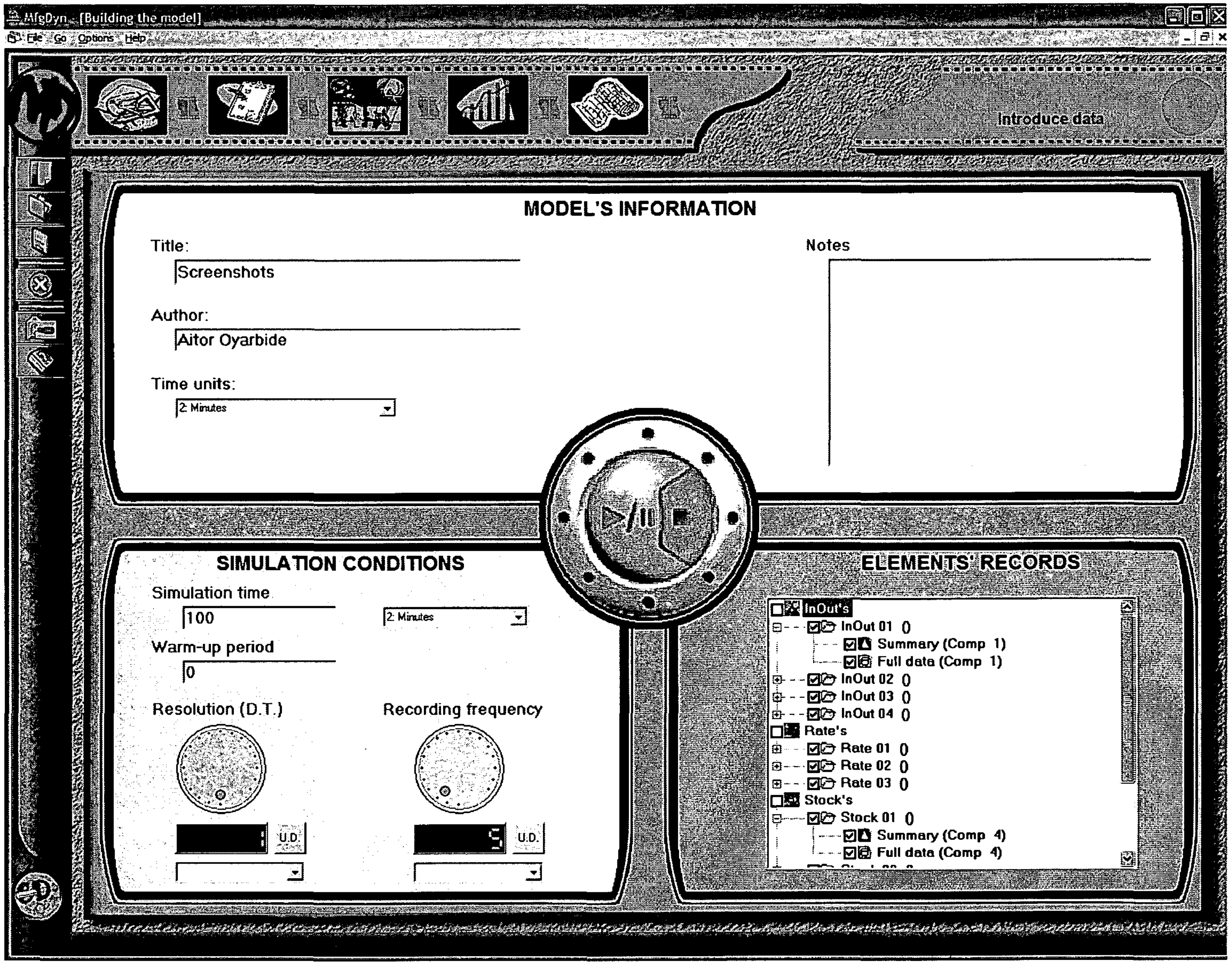
Type: 1: Single
Quantity: 1

Cadence: 0
Breakdowns (%): 0
Setups (%): 0

From: Select Source ... To: Select Taget ...
Quantity In: 1 Out: 1 (u)

Select Output ...
Select Priority ...

Figure 68: Screenshot of the 'InOut', 'Stock' and 'Rate' properties sub-form



MODEL'S INFORMATION

Title: Screenshots

Author: Aitor Oyarbide

Time units: 2: Minutes

Notes

SIMULATION CONDITIONS

Simulation time: 100 — 2: Minutes

Warm-up period: 0

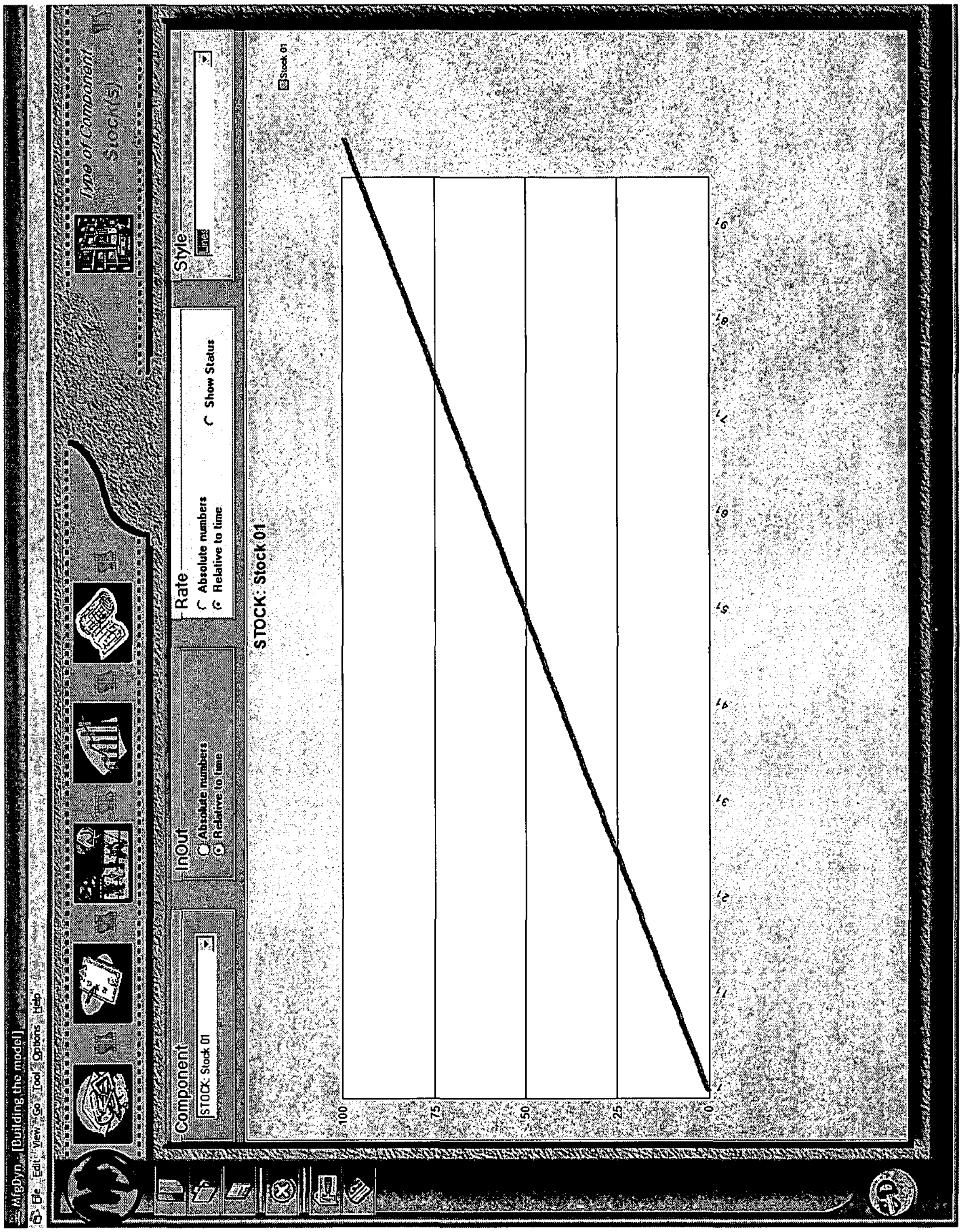
Resolution (D.T.)

Recording frequency: 5

ELEMENTS' RECORDS

- ☐ InOut's
  - ☑ InOut 01 ()
    - ☑ Summary (Comp 1)
    - ☑ Full data (Comp 1)
  - ☑ InOut 02 ()
  - ☑ InOut 03 ()
  - ☑ InOut 04 ()
- ☐ Rate's
  - ☑ Rate 01 ()
  - ☑ Rate 02 ()
  - ☑ Rate 03 ()
- ☐ Stock's
  - ☑ Stock 01 ()
    - ☑ Summary (Comp 4)
    - ☑ Full data (Comp 4)

Figure 69: Screenshot of the model execution form



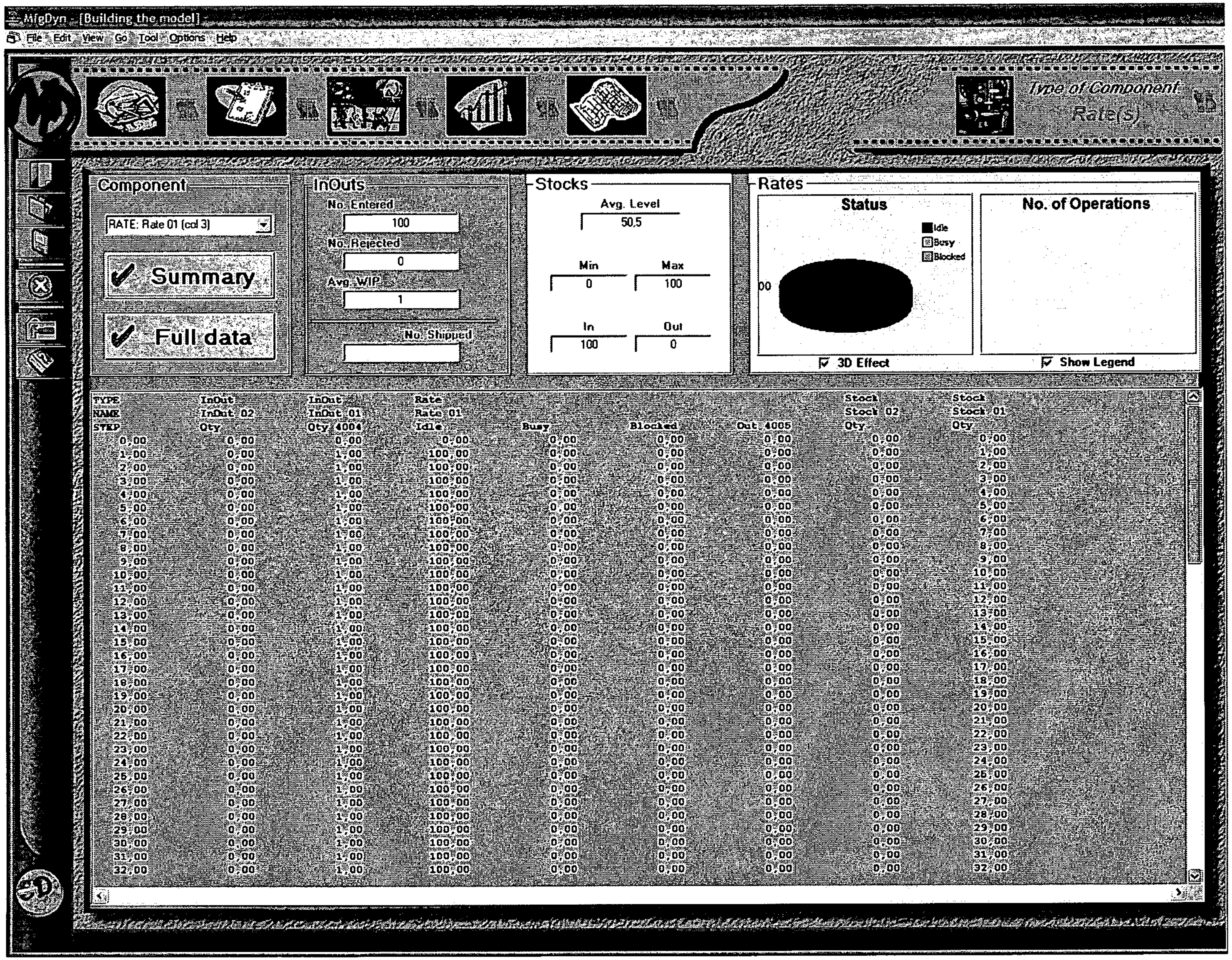
Component: STOCK: Stock 01

InOut:
○ Absolute numbers
● Relative to time

Rate:
○ Absolute numbers
● Relative to time
○ Show Status

Style: Line

Figure 70: Screenshot of the graphical results form



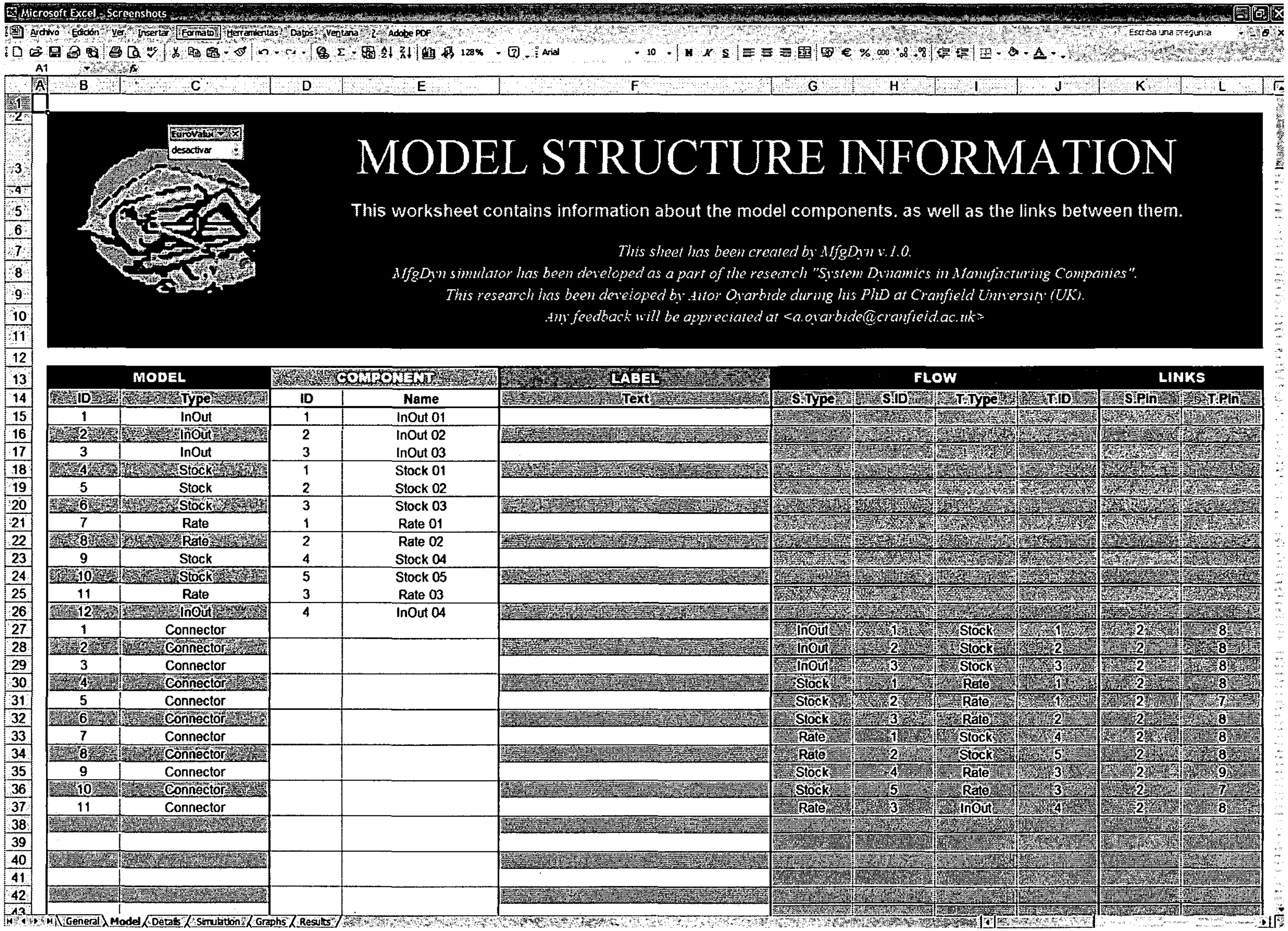
RATE: Rate 01 (col 3)

No. Entered: 100
No. Rejected: 0
Avg. WIP: 1
No. Shipped:

Avg. Level: 50,5
Min: 0
Max: 100
In: 100
Out: 0

TYPE	InOut	InOut	Rate				Stock	Stock
NAME	InOut 02	InOut 01	Rate 01				Stock 02	Stock 01
STEP	Qty	Qty 4004	Idle	Busy	Blocked	Out 4005	Qty	Qty
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	1,00
2,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	2,00
3,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	3,00
4,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	4,00
5,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	5,00
6,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	6,00
7,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	7,00
8,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	8,00
9,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	9,00
10,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	10,00
11,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	11,00
12,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	12,00
13,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	13,00
14,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	14,00
15,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	15,00
16,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	16,00
17,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	17,00
18,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	18,00
19,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	19,00
20,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	20,00
21,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	21,00
22,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	22,00
23,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	23,00
24,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	24,00
25,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	25,00
26,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	26,00
27,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	27,00
28,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	28,00
29,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	29,00
30,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	30,00
31,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	31,00
32,00	0,00	1,00	100,00	0,00	0,00	0,00	0,00	32,00

Figure 71: Screenshot of the numerical results form



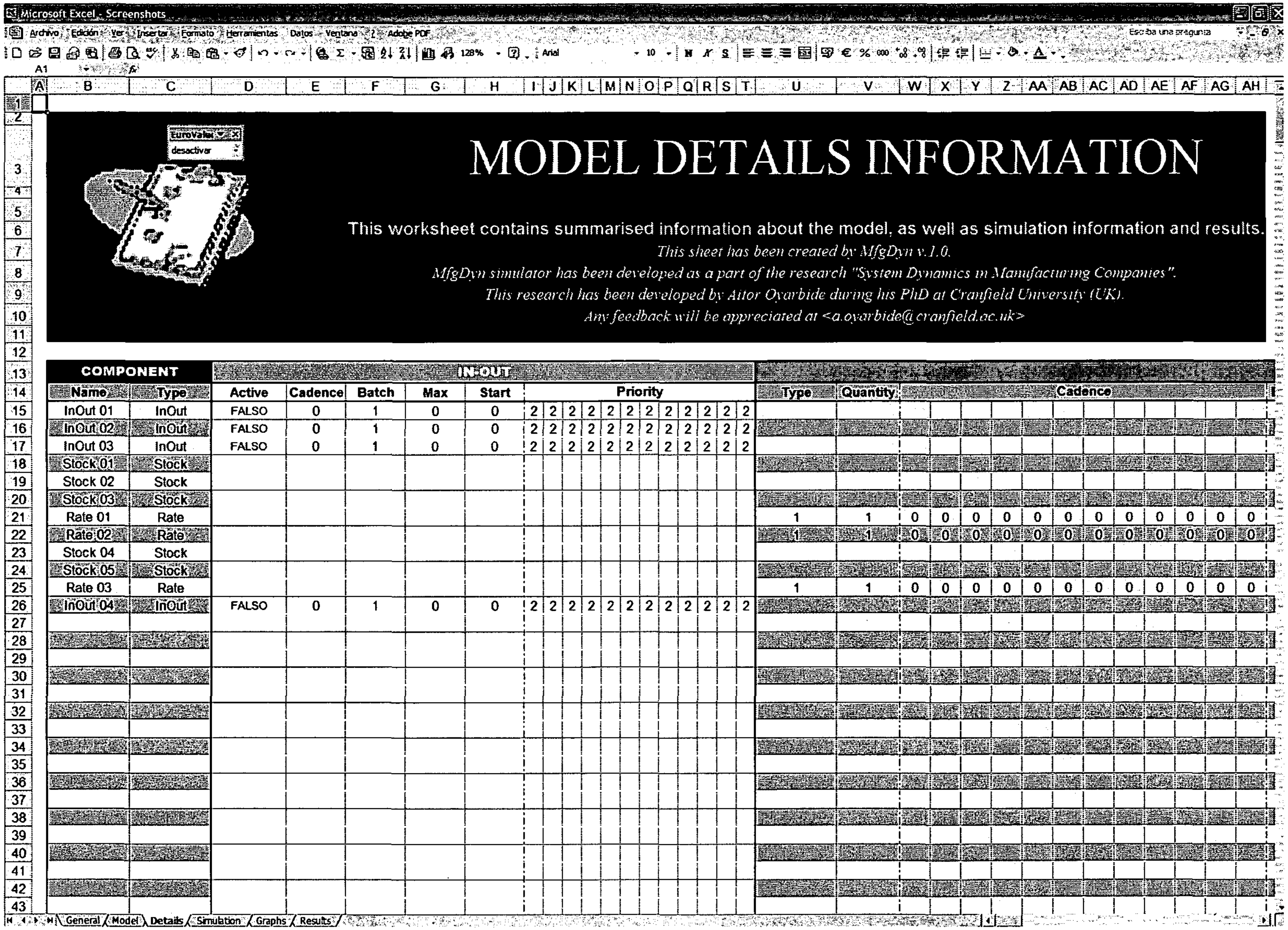
MODEL STRUCTURE INFORMATION

This worksheet contains information about the model components, as well as the links between them.

This sheet has been created by MfgDyn v.1.0.
MfgDyn simulator has been developed as a part of the research "System Dynamics in Manufacturing Companies".
This research has been developed by Aitor Oyarbide during his PhD at Cranfield University (UK).
Any feedback will be appreciated at <a.oyarbide@cranfield.ac.uk>

MODEL		COMPONENT		LABEL	FLOW				LINKS	
ID	Type	ID	Name	Text	S.Type	S.ID	T.Type	T.ID	S.Pin	T.Pin
1	InOut	1	InOut 01							
2	InOut	2	InOut 02							
3	InOut	3	InOut 03							
4	Stock	1	Stock 01							
5	Stock	2	Stock 02							
6	Stock	3	Stock 03							
7	Rate	1	Rate 01							
8	Rate	2	Rate 02							
9	Stock	4	Stock 04							
10	Stock	5	Stock 05							
11	Rate	3	Rate 03							
12	InOut	4	InOut 04							
1	Connector				InOut	1	Stock	1	2	8
2	Connector				InOut	2	Stock	2	2	8
3	Connector				InOut	3	Stock	3	2	8
4	Connector				Stock	1	Rate	1	2	8
5	Connector				Stock	2	Rate	1	2	7
6	Connector				Stock	3	Rate	2	2	8
7	Connector				Rate	1	Stock	4	2	8
8	Connector				Rate	2	Stock	5	2	8
9	Connector				Stock	4	Rate	3	2	9
10	Connector				Stock	5	Rate	3	2	7
11	Connector				Rate	3	InOut	4	2	8

Figure 72: Screenshot of the excel file that stores the model's diagram



MODEL DETAILS INFORMATION

This worksheet contains summarised information about the model, as well as simulation information and results.

This sheet has been created by MfgDyn v.1.0.

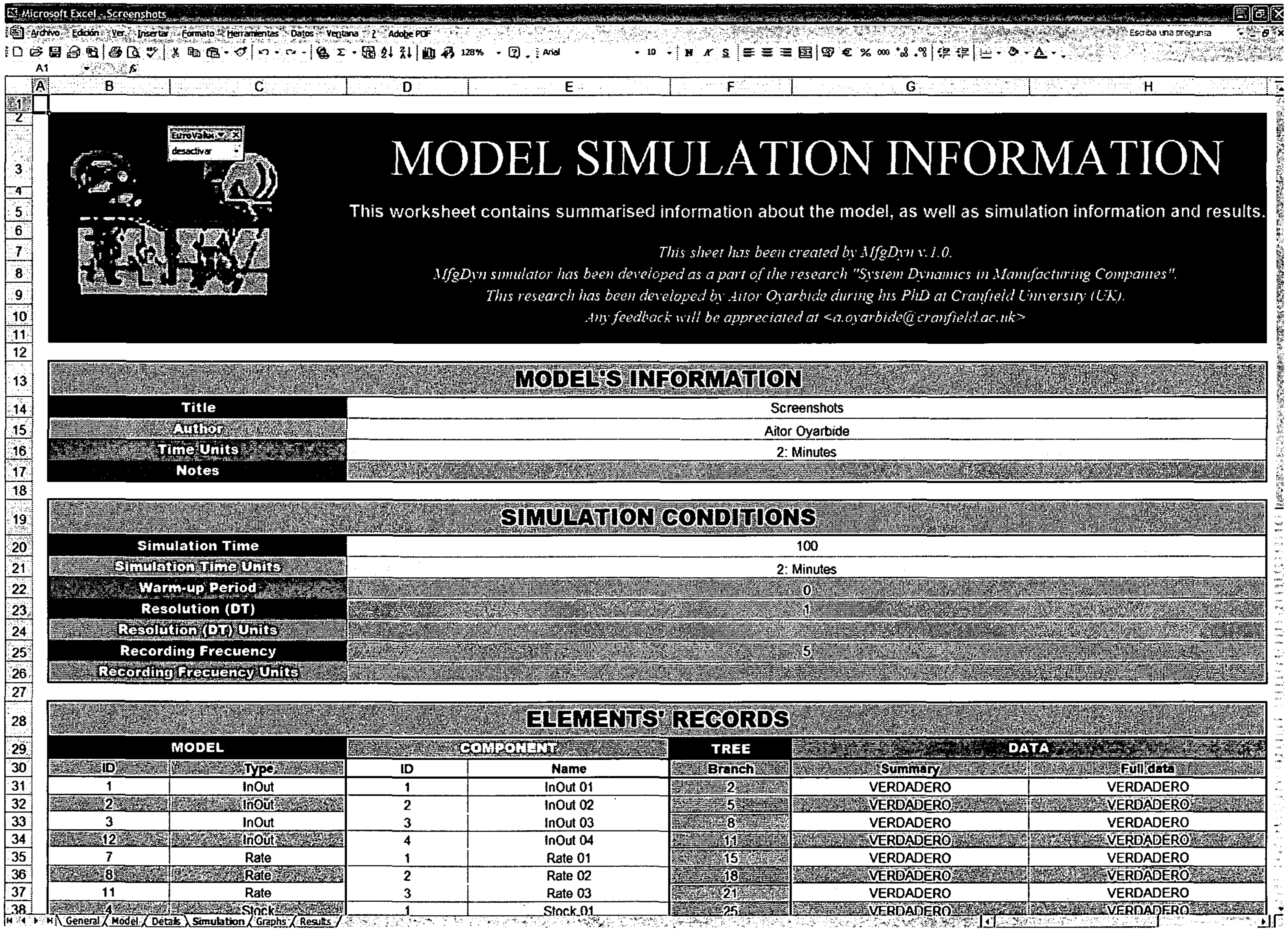
MfgDyn simulator has been developed as a part of the research "System Dynamics in Manufacturing Companies".

This research has been developed by Aitor Oyarbide during his PhD at Cranfield University (UK).

Any feedback will be appreciated at <a.oyarbide@cranfield.ac.uk>

COMPONENT		IN-OUT																														
Name	**Type**	**Active**	**Cadence**	**Batch**	**Max**	**Start**	**Priority**												**Type**	**Quantity**	**Cadence**											
InOut 01	InOut	FALSO	0	1	0	0	2	2	2	2	2	2	2	2	2	2	2	2														
InOut 02	InOut	FALSO	0	1	0	0	2	2	2	2	2	2	2	2	2	2	2	2														
InOut 03	InOut	FALSO	0	1	0	0	2	2	2	2	2	2	2	2	2	2	2	2														
Stock 01	Stock																															
Stock 02	Stock																															
Stock 03	Stock																															
Rate 01	Rate																		1	1	0	0	0	0	0	0	0	0	0	0	0	0
Rate 02	Rate																		1	1	0	0	0	0	0	0	0	0	0	0	0	0
Stock 04	Stock																															
Stock 05	Stock																															
Rate 03	Rate																		1	1	0	0	0	0	0	0	0	0	0	0	0	0
InOut 04	InOut	FALSO	0	1	0	0	2	2	2	2	2	2	2	2	2	2	2	2														

Figure 73: Screenshot of the excel file that stores the model's data



MODEL SIMULATION INFORMATION

This worksheet contains summarised information about the model, as well as simulation information and results.

This sheet has been created by MfgDyn v.1.0.
MfgDyn simulator has been developed as a part of the research "System Dynamics in Manufacturing Companies".
This research has been developed by Aitor Oyarbide during his PhD at Cranfield University (UK).
Any feedback will be appreciated at <a.oyarbide@cranfield.ac.uk>

MODEL'S INFORMATION

Title	Screenshots
Author	Aitor Oyarbide
Time Units	2: Minutes
Notes	

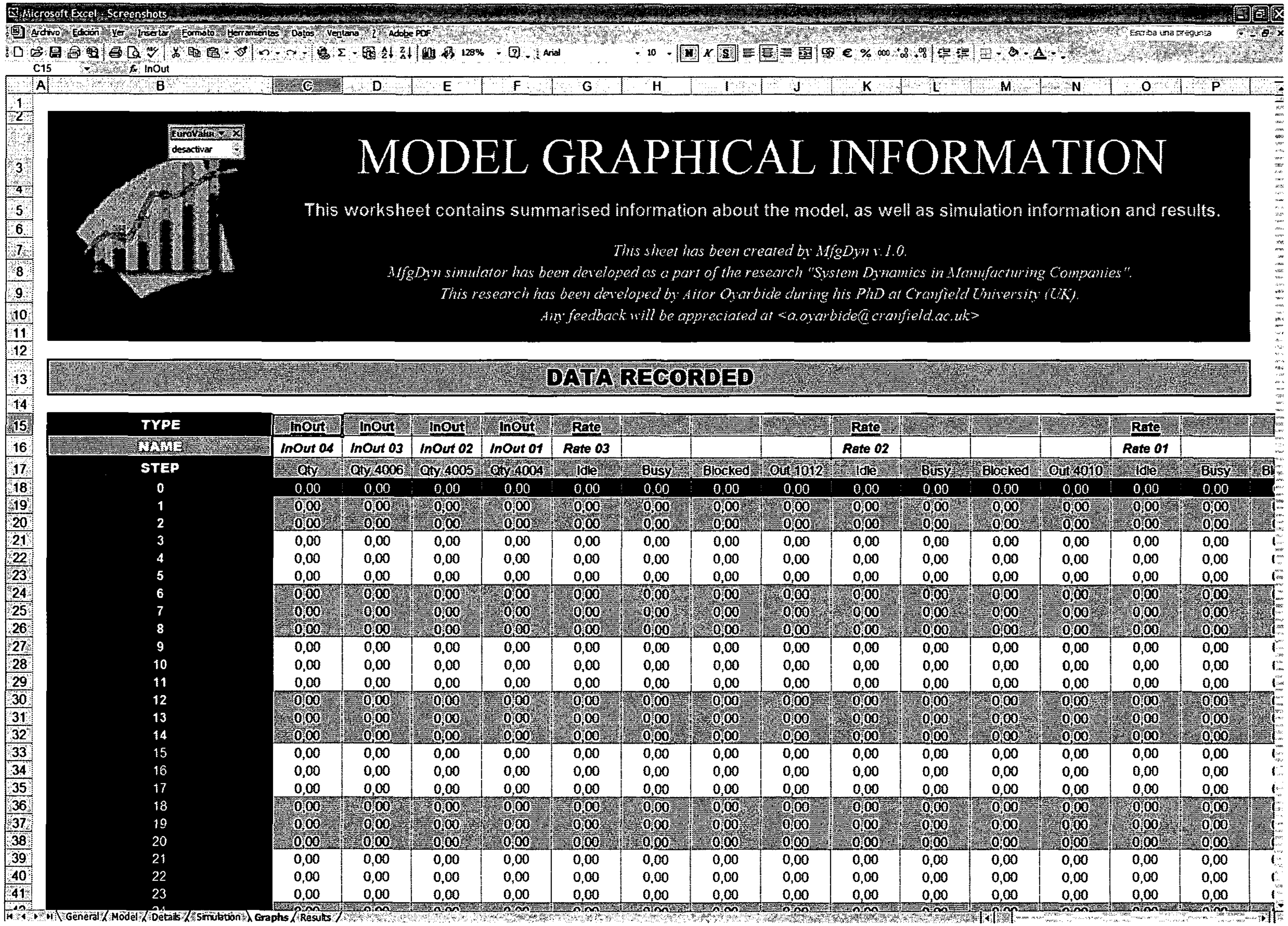
SIMULATION CONDITIONS

Simulation Time	100
Simulation Time Units	2: Minutes
Warm-up Period	0
Resolution (DT)	1
Resolution (DT) Units	
Recording Frecuency	5
Recording Frecuency Units	

ELEMENTS' RECORDS

MODEL		COMPONENT		TREE	DATA	
ID	Type	ID	Name	Branch	Summary	Full data
1	InOut	1	InOut 01	2	VERDADERO	VERDADERO
2	InOut	2	InOut 02	5	VERDADERO	VERDADERO
3	InOut	3	InOut 03	8	VERDADERO	VERDADERO
12	InOut	4	InOut 04	11	VERDADERO	VERDADERO
7	Rate	1	Rate 01	15	VERDADERO	VERDADERO
8	Rate	2	Rate 02	18	VERDADERO	VERDADERO
11	Rate	3	Rate 03	21	VERDADERO	VERDADERO
4	Stock	1	Stock 01	25	VERDADERO	VERDADERO

Figure 74: Screenshot of the excel file that stores the execution details



MODEL GRAPHICAL INFORMATION

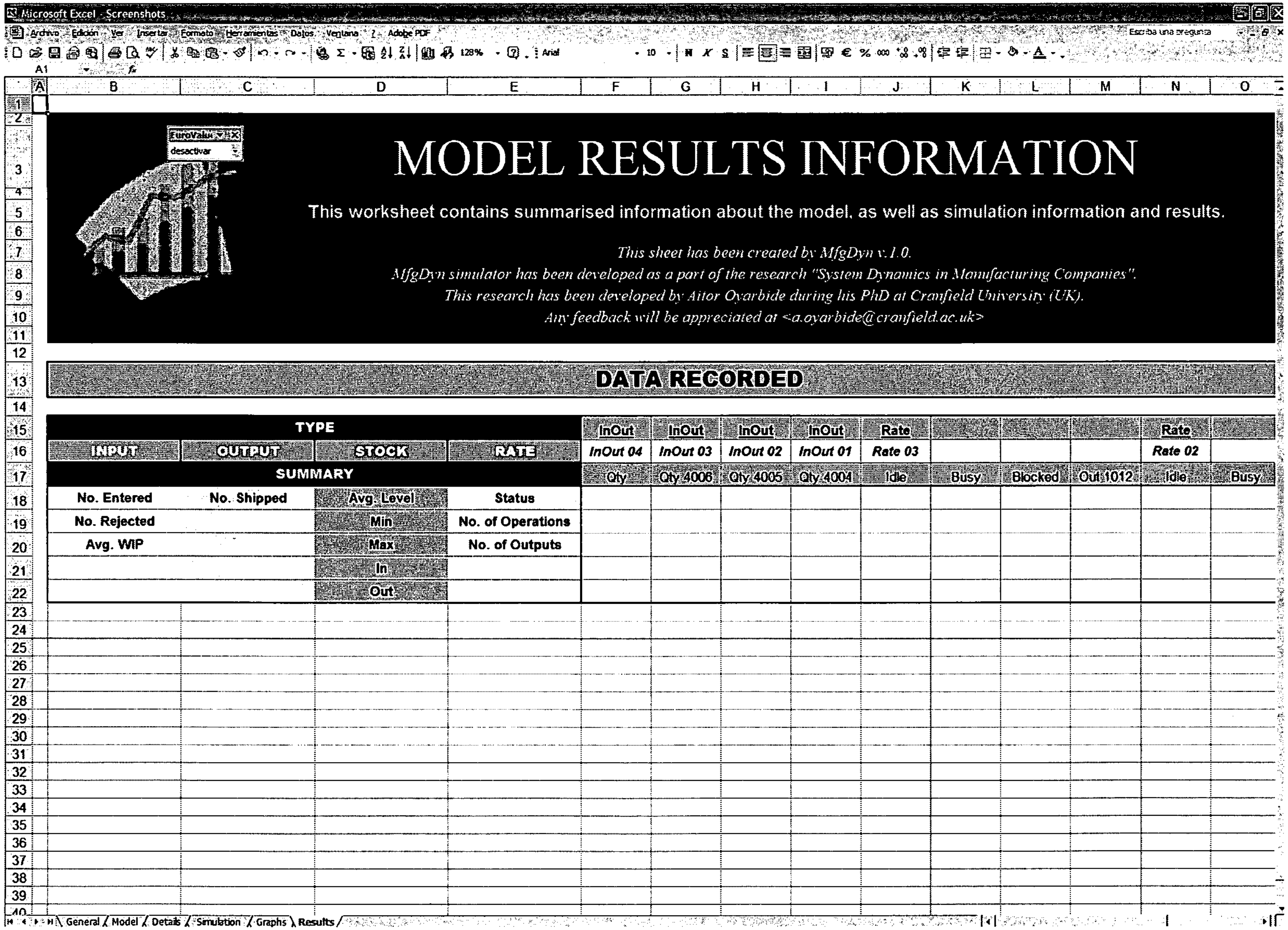
This worksheet contains summarised information about the model, as well as simulation information and results.

This sheet has been created by MfgDyn v.1.0.
MfgDyn simulator has been developed as a part of the research "System Dynamics in Manufacturing Companies".
This research has been developed by Aitor Oyarbide during his PhD at Cranfield University (UK).
Any feedback will be appreciated at <a.oyarbide@cranfield.ac.uk>

DATA RECORDED

TYPE	InOut	InOut	InOut	InOut	Rate				Rate				Rate	
NAME	InOut 04	InOut 03	InOut 02	InOut 01	Rate 03				Rate 02				Rate 01	
STEP	Qty	Qty 4006	Qty 4005	Qty 4004	Idle	Busy	Blocked	Out 1012	Idle	Busy	Blocked	Out 4010	Idle	Busy
0	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
4	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
6	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
7	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
8	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
9	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
12	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
16	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
19	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
21	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
22	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
23	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Figure 75: Screenshot of the excel file that stores the output data



MODEL RESULTS INFORMATION

This worksheet contains summarised information about the model, as well as simulation information and results.

This sheet has been created by MfgDyn v.1.0.
MfgDyn simulator has been developed as a part of the research "System Dynamics in Manufacturing Companies".
This research has been developed by Aitor Oyarbide during his PhD at Cranfield University (UK).
Any feedback will be appreciated at <a.oyarbide@cranfield.ac.uk>

DATA RECORDED

TYPE				**InOut**	**InOut**	**InOut**	**InOut**	**Rate**				**Rate**	
INPUT	**OUTPUT**	**STOCK**	**RATE**	***InOut 04***	***InOut 03***	***InOut 02***	***InOut 01***	***Rate 03***				***Rate 02***	
SUMMARY				Qty	Qty 4006	Qty 4005	Qty 4004	Idle	Busy	Blocked	Out 1012	Idle	Busy
No. Entered	**No. Shipped**	Avg. Level	**Status**										
No. Rejected		Min	**No. of Operations**										
Avg. WIP		Max	**No. of Outputs**										
		In											
		Out											

Figure 76: Screenshot of the excel file that stores the summary output data

Appendix F

CODE IMPLEMENTED IN THE ‘DESIGN’ FORM

```
Option Explicit
'#############################################################################################
'#####                                  AREA DE DECLARACIONES                            #####
'#############################################################################################

'Funcion que permite la copia de un archivo a otro (para copiar Blank.xls)
' Se utiliza: Call CopyFile ("Origen", "Destino", x) donde:
' Si x=0 (sobreescribir) y si x<>0 (no sobreescribe en caso de que el fichero exista)
Private Declare Sub CopyFile Lib "kernel32" Alias "CopyFileA" ( _
                    ByVal lpExistingFileName As String, ByVal lpNewFileName As String, _
                    ByVal bFailIfExists As Long)

'Funcion que se utiliza para imprimir la pantalla que se esta visualizando
Private Declare Sub keybd_event Lib "user32" ( _
                    ByVal bVk As Byte, ByVal bScan As Byte, ByVal dwFlags As Long, _
                    ByVal dwExtraInfo As Long)


'----------------------------------------   CONSTANTES   -------------------------------------
'Tamaño de los iconos (Tamaño icono + offset)
Const REFERENCIA_VCAL = 900 + 60
Const REFERENCIA_HTAL = 900 + 60

Const RITMO_VCAL = 1200 + 60
Const RITMO_HTAL = 900 + 60

Const ALMACEN_VCAL = 900 + 60
Const ALMACEN_HTAL = 900 + 60

Const CONVERSOR_VCAL = 600 + 60
Const CONVERSOR_HTAL = 600 + 60

Const CONSTANTE_VCAL = 600 + 60
Const CONSTANTE_HTAL = 600 + 60

Const BORDE_OFFSET = 30                             'Margen para el recuadro de selección
Const CONECTOR_OFFSET = 100                         'La mitad del tamaño del circulo
Const FOTOGRAMAS_CONTRAFILM = 20

'Movimiento grande y pequeño del scroll
Const SCROLL_GRANDE = 350
Const SCROLL_PEQUEÑO = 150


'----------------------------------------   VARIABLES   ---------------------------------------
Private mbRejillaOcupada(1 To 100, 1 To 100) As Boolean 'Rejilla de celdas ocupadas

Private msBtnActivo As String                       'Estilo del boton seleccionado
Private miPosX, miPosY As Single                    'Posicion del raton

Private mbConectorEnCurso As Boolean                'Indica si se está poniendo un conector
Private mbConexionesDisponibles(1 To 12) As Boolean 'Links que se encuentran disponibles
Private miConectorOptimo As Integer

'Indica que se ha clickado y no se debe mover
Private mbConectorTmp1Fijado As Boolean
Private mbConectorTmp2Fijado As Boolean

'Indica en que componense se coloca el conector
Private miTipoOrg As String                         'Tipo de componente
Private miTipoFin As String
Private miConectorOrgComponente As Integer          'ID de componente
Private miConectorFinComponente As Integer
Private miPinUtilizadoOrg As Integer                'Pin de componente
Private miPinUtilizadoFin As Integer

Private mbVerProceso As Boolean                     'Indica si se esta viendo el flujo
Private miEtiquetaSeleccionada As Integer           'Indica si una etiqueta ha sido seleccionad
```

```
'#############################################################################################
'#####                        FORMULARIO EN GENERAL                                      #####
'#############################################################################################


'-InicializarFormulario-----------------------------------------------------------------------
' OBJETO > Todos en blanco y negro menos el formulario actual
'---------------------------------------------------------------------------------------------
Private Sub InicializarFormulario(Tipo_IN As String)
  gsFormActivo = "Diseño"                         'Se introduce el form cargado
  mbVerProceso = True                             'Se ven las "flechas"

  imgContrafilm.Visible = False                   'Oculta contrafilm
  fwlCargando.Visible = False                     'Oculta mensaje de carga de ficheros
  fwlGrabando.Visible = False

  Call InicializarPelicula                        'Activa el boton de Diseño
  Call InicializarBotoneraDiseño                  'Activa botones prefijados
  Call ActivarBordeBoton("Puntero")               'Preseleccionamos "Puntero", lo rebordeamos
  If Tipo_IN = "Activar" Then Call CrearElementos 'Solo se crean elementos y se redibuja
  If Tipo_IN = "Activar" Then Call RedibujarModelo  'cuando volvemos de otras pantallas
  Call InicializarAreaVirtual                     'Coloca los scrolls

  If gbInicioAbrir = True Then Call mnuFicheroAbrir_Click 'Se quiere abrir un modelo

  lblEstadComp.Caption = MSG_ESTAD_COMP + Str(giCompNum)     'Actualiza el contrafilm
  lblEstadFluj.Caption = MSG_ESTAD_FLUJ + Str(giFlujNum)
  lblEstadEtiq.Caption = MSG_ESTAD_ETIQ + Str(giEtiqNum)
  lblEstadComp.Visible = False                    'Oculta el contrafilm
  lblEstadFluj.Visible = False
  lblEstadEtiq.Visible = False

  lblRatonXY.Visible = False                      'Oculta la posicion del raton

  'Animar el ContraFilm; Activar el timer (Parar, 100ms, activar)
  tmrContrafilm.Enabled = False
  tmrContrafilm.Interval = 50
  tmrContrafilm.Enabled = True
End Sub


'-mnuFicheroSalir_Click, imgSalir_Click-------------------------------------------------------
' OBJETO > Sale de la aplicacion
'---------------------------------------------------------------------------------------------
Private Sub mnuFicheroSalir_Click()
  On Error Resume Next

  'Si el modelo actual ha sido modificado y no se desean perder los cambios -> Salir
  If gbModeloModificado = True And giCompNum <> 0 Then
    If MsgBox(MSG_PERDER_CAMBIOS, vbYesNo + vbCritical + _
              vbDefaultButton2, MSG_ABANDONAR_APLICACION) = vbNo Then
        Exit Sub
    End If
  End If

  'DESCARGAR EL FICHERO EXCEL DE LA MEMORIA       'Por si algo se ha quedado abierto
  oXlsApp.Quit
  Set oXlsPag = Nothing
  Set oXlsLib = Nothing
  Set oXlsApp = Nothing

  Unload frmDiseño                                'Descarga formularios
  Unload frmPrincipal
  End                                             'Cierra la aplicacion DEL TODO
End Sub


Private Sub imgSalir_Click()
  Call mnuFicheroSalir_Click
End Sub



'#############################################################################################
'#####                        FILM                                                       #####
'#############################################################################################


'-mnuIrDiseño_Click, ..., mnuIrInforme_Click, imgDetalle_Click, ..., imgInforme_Click--------
' OBJETO > Mostrar el formulario correspondiente (en el actual no se hace nada)
'---------------------------------------------------------------------------------------------
Private Sub mnuIrDiseño_Click()
  Exit Sub
End Sub
```

```
'#############################################################################################
'#####                         AREA VIRTUAL                                              #####
'#############################################################################################


'-sbrHorizontal_Change,sbrVertical_Change-----------------------------------------------------
' OBJETO > Actualizar el area de trabajo cuando se mueve el scroll
'---------------------------------------------------------------------------------------------
Private Sub sbrHorizontal_Change()
  fraAreaVirtual.Left = -sbrHorizontal.Value       'Mueve el area virtual
End Sub


Private Sub sbrVertical_Change()
  fraAreaVirtual.Top = -sbrVertical.Value
End Sub



'#############################################################################################
'#####                         MENSAJES                                                  #####
'#############################################################################################


'-MsgEstado-----------------------------------------------------------------------------------
' OBJETO > Mostrar el mensaje de estado de las pantallas
'---------------------------------------------------------------------------------------------
Private Sub MsgEstado(Msg_IN As String)
  Const MSG_TEMPORIZADOR = 3000                    '3000 milisegundos

  tmrMsgEstado.Enabled = False                     'Para el temporizador
  lblMsgEstado.Caption = Msg_IN                    'Carga el mensaje

  If Msg_IN <> MSG_VACIO Then
    tmrMsgEstado.Interval = MSG_TEMPORIZADOR       'Carga el tiempo predefinado
    tmrMsgEstado.Enabled = True                    'Activa el temporizador
  End If
End Sub


'-tmrMsgEstado_Timer--------------------------------------------------------------------------
' OBJETO > Borrar el mensaje de estado
'---------------------------------------------------------------------------------------------
Private Sub tmrMsgEstado_Timer()
  Call MsgEstado(MSG_VACIO)
End Sub

'-picComp_Click-------------------------------------------------------------------------------
' OBJETO > Seleccionar/deseleccionar iconos
'---------------------------------------------------------------------------------------------
Private Sub picComp_Click(Indice_IN As Integer)
  Dim iI As Integer

  Select Case msBtnActivo
    Case "Puntero"
      'Si hay un componente seleccionado y es el que hemos pulsado
      If shpSeleccion.Visible = True And shpSeleccion.Tag = txtComp(Indice_IN).Text Then
        'Deseleccionar el componente
        gvComp(Indice_IN).Seleccionado = False     'Lo deselecciona
        shpSeleccion.Visible = False               'Oculta el recuadro
      Else
        For iI = 1 To giEtiqNum                    'Deseleccionar todas las etiquetas
          lblEtiq(iI).ForeColor = COLOR_MORADO     'Color original
        Next iI
        miEtiquetaSeleccionada = 0                 'Deselecciona la etiqueta

        'Seleccionar el componente pulsado
        'Como no podemos saber cual estaba seleccionado, los borramos todos
        For iI = 1 To giCompNum
          gvComp(iI).Seleccionado = False
        Next iI

        Call AjustarBorde(Indice_IN)               'Tamaño y la posición del borde
        shpSeleccion.Tag = txtComp(Indice_IN).Text 'Carga su nombre
        gvComp(Indice_IN).Seleccionado = True      'Lo selecciona
        shpSeleccion.Visible = True                'Lo visualiza
        Beep
      End If
    Case "Conector"                                'Insertar nuevo flujo
      If mbConectorEnCurso = False Then            'No hay ningun conector empezado
        mbConectorEnCurso = True
        Call MsgEstado(MSG_CONECTOR_DESTINO)
        mbConectorTmp1Fijado = True
      Else                                         'Hay un conector en curso
```

```
        mbConectorEnCurso = False
        mbConectorTmplFijado = False

        giFlujNum = giFlujNum + 1                  'Incrementa el número de conexiones
        lblEstadFluj.Caption = MSG_ESTAD_FLUJ + Str(giFlujNum)

        shpFlujoTmp1.Visible = False               'Oculta los conectores temporales
        shpFlujoTmp2.Visible = False

        'Crea el nuevo flujo a nivel del área de trabajo
        Load shpFlujoOrg(giFlujNum)
        Load shpFlujoFin(giFlujNum)

        shpFlujoOrg(giFlujNum).Left = shpFlujoTmp1.Left
        shpFlujoOrg(giFlujNum).Top = shpFlujoTmp1.Top
        shpFlujoOrg(giFlujNum).Visible = True

        shpFlujoFin(giFlujNum).Left = shpFlujoTmp2.Left
        shpFlujoFin(giFlujNum).Top = shpFlujoTmp2.Top
        shpFlujoFin(giFlujNum).Visible = True

        'Crea el nuevo flujo a nivel de tabla
        gvFluj(giFlujNum).ModID = giFlujNum
        gvFluj(giFlujNum).TipoOrg = miTipoOrg
        gvFluj(giFlujNum).TipoFin = miTipoFin
        gvFluj(giFlujNum).IDOrg = miConectorOrgComponente
        gvFluj(giFlujNum).PinOrg = miPinUtilizadoOrg
        gvFluj(giFlujNum).IDFin = miConectorFinComponente
        gvFluj(giFlujNum).PinFin = miPinUtilizadoFin
        gvFluj(giFlujNum).ConOrgIzda = shpFlujoTmp1.Left
        gvFluj(giFlujNum).ConOrgArriba = shpFlujoTmp1.Top
        gvFluj(giFlujNum).ConFinIzda = shpFlujoTmp2.Left
        gvFluj(giFlujNum).ConFinArriba = shpFlujoTmp2.Top

        Call DibujarFlujo("Nuevo", (giFlujNum))
      End If
  End Select
End Sub

'##############################################################################################
'#####                              AREA DE TRABAJO                                     #####
'##############################################################################################

'-fraAreaVirtual_MouseDown---------------------------------------------------------------------
' OBJETO > Mirar si hay espacio para colocar un componente y colocarlo si es posible
'----------------------------------------------------------------------------------------------
Private Sub fraAreaVirtual_MouseDown(Btn_IN As Integer, Sft_IN As Integer, _
                                     X_IN As Single, Y_IN As Single)
  Dim iI As Integer

  Select Case msBtnActivo                          'Las acciones son diferentes segun el boton
    Case "Puntero"
      'Deseleccionar todos los componentes
      For iI = 1 To giCompNum
        gvComp(iI).Seleccionado = False            'Lo deselecciona
        shpSeleccion.Visible = False               'Oculta el recuadro
      Next iI
    Case "Conector"
      Exit Sub
    Case "InOut", "Rate", "Stock", "Converter", "Constant"
      Call InsertarNuevoComponente(msBtnActivo, X_IN, Y_IN)
    Case "Etiqueta"
      If miEtiquetaSeleccionada = 0 Or giEtiqNum = 0 Then
        Call InsertarNuevaEtiqueta(X_IN, Y_IN)
      Else
        Call MoverEtiqueta(miEtiquetaSeleccionada, X_IN, Y_IN)
        lblEtiq(miEtiquetaSeleccionada).ForeColor = COLOR_MORADO 'Deseleccionarlo
        miEtiquetaSeleccionada = 0
      End If
  End Select
End Sub
```

Appendix G

CASE STUDY – PRESENTATION LETTER

{Date}
{Sr/Srta}
{Job title}
{Company name}, {Address}

Análisis de procesos productivos mediante simulación continua

Estimado {Name}

Mi nombre es Aitor Oyarbide y le escribo desde la Universidad de Cranfield donde estoy efectuando un proyecto de investigación. Tu {Company} ha llamado nuestra atención, a través de una conversación mantenida con un colega de la universidad, que ha realizado previamente proyectos en su empresa. Nuestro trabajo está relacionado con la simulación de procesos, y más concretamente, con la aplicación de métodos de simulación continua como forma de ganar capacidad competitiva. Tanto si su empresa emplea la simulación como si no, este proyecto puede ser beneficial para ustedes, y facilitar el diseño de nuevos procesos o el rediseño de procesos existentes, y es por eso que nos gustaría colaborar con ustedes.

El estado actual del proyecto requiere que visitemos y analicemos 4 procesos de otras tantas empresas. Es por eso que estoy interesado en su compañía, y pienso que podría añadir un gran valor a nuestro proyecto. Por lo tanto, me gustaría preguntarle si podríamos realizar dicho experimento en {Company name}.

Déjeme explicarle en que consiste el experimento: Una primera visita analizará la idoneidad de la empresa/proceso y determinará el subproceso a simular. Una segunda visita realizará la recogida de datos para su posterior simulación. Finalmente, una visita (si es de interés para ustedes) les mostrará los modelos realizados y las conclusiones de los mismos. Este experimente es desarrollado únicamente con fines académicos y toda la información (así como el nombre de la empresa) serán guardados confidencialmente.

Para {Company name} la participación en este proyecto será una oportunidad para aprender con nosotros, y evaluar las potencialidades de formas alternativas de simulación. Espero que usted esté interesado en participar en este proyecto innovador por lo cual recibirá una llamada mía en breve. Mientras tanto, si dispone de cuestiones no dude en contactar conmigo en el telefono 943 794700 o por email en a.oyarbide@cranfield.ac.uk.

Sin más, le Saluda atentamente,

Aitor Oyarbide

School of Industrial and Manufacturing Science

Cranfield University, Cranfield, MK43 0AL, UK.