A Dynamic Simulation Framework for Biopharmaceutical Capacity Management

A thesis submitted to University College London

for the degree of

DOCTOR OF ENGINEERING

by

Paige Ashouri, *MEng*

The Advanced Centre for Biochemical Engineering

Department of Biochemical Engineering

University College London

Torrington Place

London WC1E 7JE

United Kingdom

January 2011

I, Paige Ashouri, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

.....

To my parents with love and gratitude

ABSTRACT

In biopharmaceutical manufacturing there have been significant increases in drug complexity, risk of clinical failure, regulatory pressures and demand. Compounded with the rise in competition and pressures of maintaining high profit margins this means that manufacturers have to produce more efficient and lower capital intensive processes. More are opting to use simulation tools to perform such revisions and to experiment with various process alternatives, activities which would be time consuming and expensive to carry out within the real system.

A review of existing models created for different biopharmaceutical activities using the Extend® (ImagineThat!, CA) platform led to the development of a standard framework to guide the design and construct of a more efficient model. The premise of the framework was that any 'good' model should meet five requirement specifications: 1) Intuitive to the user, 2) Short Run-Time, 3) Short Development Time, 4) Relevant and has Ease of Data Input/Output, and 5) Maximised Reusability and Sustainability. Three different case studies were used to test the framework, two biotechnology manufacturing and one fill/finish, with each adding a new layer of understanding and depth to the standard due to the challenges faced. These Included procedures and constraints related to complex resource allocation, multi-product scheduling and complex 'lookahead' logic for scheduling activities such as buffer makeup and difficulties surrounding data availability. Subsequently, in order to review the relevance of the models, various analyses were carried out including schedule optimisation, debottlenecking and Monte Carlo simulations, using various data representation tools to deterministically and stochastically answer the different questions within each case study scope.

The work in this thesis demonstrated the benefits of using the developed standard as an aid to building decision-making tools for biopharmaceutical manufacturing capacity management, so as to increase the quality and efficiency of decision making to produce less capital intensive processes.

ACKNOWLEDGMENTS

I would like to thank a number of individuals who contributed to bringing this research into fruition. I would first like to express my sincere gratitude to my supervisor Dr. Suzanne Farid, for her continued guidance, encouragement and patience throughout the course of this research. Her advice and enthusiasm proved invaluable to my progress. I would also like to thank Professor Nigel Titchener-Hooker for his invaluable advice and for offering me the opportunity to do this EngD.

I am also grateful to my EngD sponsor (Eli Lilly & Co) for their financial support and my industrial supervisor Roger L Scott for his continued support, advice and motivation. I would also like to acknowledge the help and expert advice of Bernard McGarvey, Richard Dargatz and Guillermo Miroquesada, their input is kindly appreciated.

I would also like to express my appreciation to the many friends and colleagues at UCL for their encouragement and collaboration and for providing a stimulating and fun environment in which to learn and develop.

Lastly, and most importantly, I would like to thank my family. My mother Vida and father Jon, for their unconditional love, support, inspiration and patience. My brother and sister for helping to raise me and for ensuring both feet stayed firmly on the ground. To them I dedicate this thesis.

CONTENTS

LIST OF TABLES

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Appendix C

LIST OF FIGURES

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Appendix B

CHAPTER 1

Scope and Background

1.1 Introduction

The philosophy of biopharmaceutical manufacture has changed over the years shifting from a heavy focus on accelerating time to market to improving process economics (Farid, 2009). This shift can be attributed to the continuously changing trends in biopharmaceuticals with the likelihood of finding blockbuster indications having diminished greatly over the last decade due to increased competition, dosage reductions and diagnostic improvements (Jagschies, 2008). The lack of blockbuster scale market share and the resulting high profit margins compounded with problems such as increased drug complexity and risk of clinical failure (Berg et al, 2005), regulatory pressures and increasing demand (Ransohoff, 2009) means that manufacturers are having to produce more efficient and lower capital intensive processes as early on in the development stages as possible.

Such requirements call for design or revision procedures which may be required in the early stages of process development (preferably) or at the height of operational activities but will always incur some degree of costs. Consequently, some biopharmaceutical manufacturers are seeking to use simulation tools to assess such revisions and to experiment with various process alternatives, activities which could be time consuming and expensive to carry out in real life experiments. With the adoption of such simulation techniques has arisen some issues as discussed by Mclean and Leong (2005); for example in the semiconductor industry they state that due to the fact that each simulation software vendor offers its own unique approach to modelling (i.e. data and graphical formats), the non-existence of standards increases the difficulties surrounding the simulation process. This particularly applies to the biopharmaceutical industry, a relative newcomer to the use of simulation

methods. The use of process simulation software in the biopharmaceutical industry is less developed than in other industries (Lim 2004, Johnston 2010). In order for these simulation techniques to add value as decision-support tools it is therefore necessary to standardize the way in which the modelling process is approached. The work in this thesis will propose such a standard.

The purpose of this chapter is to provide an overview of biopharmaceutical manufacturing and to review some of the key issues necessitating simulation. Section 1.2 highlights some of the problems faced by today's biopharmaceutical companies in greater detail and the role played by general simulation techniques while Section 1.3 highlights the complexities inherent to biopharmaceutical manufacturing. Section 1.4 discusses the aims and objectives of this thesis and Section 1.5 presents its outline.

1.2 Current Issues Facing Biopharmaceutical Companies

Figures for the commercialisation of drugs differ somewhat between information sources however all agree that the sums are enormous and growing. Less than a decade ago published figures stood at 7-12 years to take a drug from discovery to market with investments totalling $802 million (DiMasi, 2003). More recent sources state this figure now stands closer to $1.3 billion (Collier, 2009). Of these drugs 80-85% fail somewhere in the development pipeline (Polastro, 1996). The implication of this costly and risky business environment is that it is necessary for key decisions relating to process design and capacity management to be made with the goal of creating cost-effective bioprocesses that enable material to be supplied to the clinic and market on time, thus scheduling of activities should be such that demands are met with minimal delay, particularly when delay penalties are in place.

Furthermore, uncertainties in biopharmaceutical manufacture are one feature of the industry that set it apart from the more traditional pharmaceutical or the semiconductor industries (Johnston, 2010). The level of variability such as in titre, yields and cycle times means that often it is difficult to fully understand the overall impact that even minor changes or events can have. For example, the purification

requirements in terms of membrane size, with fluctuating upstream titres or the effects of random buffer equipment breakdowns on the overall process.

Regulatory constraints, particularly in cases where live organisms are used, require that additional precautions are taken with respect to facilities and equipment, such as the use of dedicated facilities and equipment, production on a campaign basis and the use of closed systems. These present a challenge with regards to facility design and process scheduling as manufacturers must overcome the complexities of area restrictions and shared resources whilst maintaining optimal scheduling. The best example of this is in the case of multi-product facilities (which are discussed further in section 1.3.1) whereby due to capacity constraints a manufacturer may have to use the same downstream processing stream for all products. Regulatory guidelines require that turnaround activities take place between different product campaigns which comprise of various shutdown and cleaning procedures. Depending on the nature of the products and the adjacent campaigns different turnaround procedures may be needed with varying cycle times and labour requirements thus giving rise to a level of complexity in process scheduling in order to minimise cost of production while meeting demand on time.

1.3 Capacity

In the 1980s more and more companies were globally investing heavily in R&D and the 1990s saw a significant overall investment boost in the biopharmaceutical industry. Too many drug failures however left the market all revved up with nowhere to go (Roth, 2001).

At the turn of the millennium however there were a large number of successes, with 65 biopharmaceuticals having gained approval between 2000 and 2004, 64 of which were protein based (Walsh, 2005). Publications of the time stated that due to the fact that many new drugs on the market required such high volumes of bio-manufacturing capacity, there was little available capacity remaining worldwide (Fox, 2001). With a significant proportion being biotechnology therapeutics such as monoclonal antibodies, the size of the recommended dose and the chronic nature of the diseases they treated meant that the volumes required were very large. As figures in 2002

showed the global cell-culture capacity seemed to be almost fully utilised (Mallik, 2002) and the industry was facing a major drought of manufacturing capacity (Thiel, 2004), with 35% of bio manufacturers outsourcing at least some of their biologics production in 2004 (Langer, 2004).

Further concerns were that not only was demand for manufacturing space already exceeding supply but the problem was expected to be at its worst in 2005(Connolly, 2001). In 2000 Immunex Corp. was forced to scale back production of its drug Enbrel due to a shortage of capacity with an estimated loss of over $200 million in revenue in 2001 (Mallik, 2002). In 2002 Roche and Trimeris announced that they would be unable to meet demand for their new HIV treatment drug (an injectable fusion inhibitor) and would have to ration supply following market launch (Fonendo, 2002).

Due to what seemed like a worldwide crisis in capacity big pharmaceutical firms went on to invest heavily in production facilities simply to meet the anticipated demands. While some companies were investing heavily in new in-house facilities, others were relying more on their relationships with contract manufacturers. Today there remains exceptionally high numbers of drugs and biologics in pharmaceutical pipelines however the capacity crunch expected to hit pharmaceutical manufacture in 2005 did not occur, partly due to the investments in capacity increases but largely due to the development of higher yielding cell systems which meant that lower volumes could yield higher product concentrations. The Second Annual Survey of Biopharmaceutical Manufacturing Capacity carried out in late 2003 indicated that the single most significant barrier to biopharmaceutical manufacture was financial while inadequacy of capacity was one of many less significant issues, falling within the same category as inadequate availability of skilled personnel.

In recent years however, factors such as the inability to find experienced staff have fallen and a more recent survey carried out put physical capacity higher up on the list of manufacturing constraints as industry titres began to increase. It is now expected that contract manufacturers will increase capacity of mammalian cell culture facilities by 91% by the year 2012, while for the biopharmaceutical developers, this figure is 43% (Langer, 2008).

As well as the costs associated with penalties (when demands are not met), market share losses or losses in potential revenues and outsourcing activities, the cost implications of creating capacity must also be taken into consideration. According to Kamarck (2006), the cost of building a new facility could be in the region of $200-$400 million while in 2005 Lonza extended its clinical-scale mammalian manufacturing capacity at a cost of £6.1 million (Winder, 2005). It could be argued that additional capacity could be built at a later stage with relatively low cost implications. However there are consequences of over- or underestimating capacity. In a study of a hypothetical monoclonal antibody product for an oncology indication, the cost to a company of 50% under-utilization of capacity was compared to the cost of a 50% shortage of capacity. The result showed that the estimated carrying cost of a facility operating at 50% capacity was $2-3million/month (based on a 500kg/yr facility) compared to an estimated loss of $40-50million/month in operating profit (Ransohoff, 2004). Thus it is necessary to not only be able to project quite accurately the future requirements of a facility but to also plan and schedule the production activities in order to maximise the capacity utilisation and thus maximise the facility's cost effectiveness.

1.3.1 Multi-product Facilities

The increase in the number of drugs in manufacture and the associated capacity constraints has meant an increase in the number of multi-product facilities (Lakhdar, 2005) of which one may consider there being two kinds. The first being whereby different products are made in parallel, simultaneously and in different production suites, and the other whereby different products are made on a campaign basis in the same part of the production facility (provided no bacterial/mammalian cell culture mix).

In order to understand the nature of such facilities one can look at contract manufacturers, the biggest adopters of multiproduct manufacturing due to the nature of their diverse product portfolios; in 2007 the global market for biopharmaceutical contract manufacturing had reached $2.4 billion, a 14% increase on the previous year and a growth expected to continue for the following year (Downey, 2008). As more and more companies are employing outside help, contractors have to cope with the

varying operations due to the variations in product demands and requirements. Some plants may use dozens of equipment to produce several different types of products leading to a myriad of ways in which a plant can be operated and finding the best operating plan and schedule has become a challenge. In order to optimise productivity in multi-product facilities where processes are run in parallel requires effective scheduling of batch overlap and efficient utilisation of shared resources (Gosling, 2003). These issues are not isolated to contract manufacturers but apply to all multiproduct scenarios. The correct design and scheduling of such processes is therefore vital to the production optimisation of such facilities. Simulation techniques can be a cost-effective way of achieving this.

1.4 Simulation Modelling in Biopharmaceutical Manufacturing

There is a growing need for simulation tools to be used during process development and manufacturing so as to minimise the potential of business losses (e.g. Farid et al 2009, Petrides 2002). Potential applications of simulation tools in biopharmaceutical manufacturing are discussed below and summarised in Figure 1.1.

1) During the idea generation stages, of both the drug and the process, project selection based on initial feasibility and economic analyses is required

2) Design of the manufacturing process and facility should be initially carried out virtually i.e. via simulations, so that any losses due to product failure at any stage during development may be minimised.

3) Scale-up from pilot plant and technology transfer typically also require further consideration of capital required, capacity utilisation and site selection; such decisions can be facilitated by simulation and optimisation studies.

4) During actual large scale manufacturing continuous process optimisation in terms of scheduling, resource utilisation and debottlenecking is required to maintain process and plant efficiency.

5) More generally, drug discovery and development times impact on the patent protection period of the commercialised drug and on the potential market figures due to changing markets and competition. Simulation techniques aid in speeding up the drug commercialisation time by allowing for quicker and more efficient process design and specification without the need for real cost and time consuming experimentations.

6) Initiatives such as Six Sigma are highlighting the importance of simulation modelling in evaluation and optimisation across the whole spectrum of pharmaceutical activities. Such activities include transactional flow and supply chain which includes transportation and logistics, procurement and supply, and marketing.

Figure 1.1 shows how simulation techniques can be used in the development stages. While manufacturing at both development and commercialisation stages remains a significant area for the use of decision support tools,

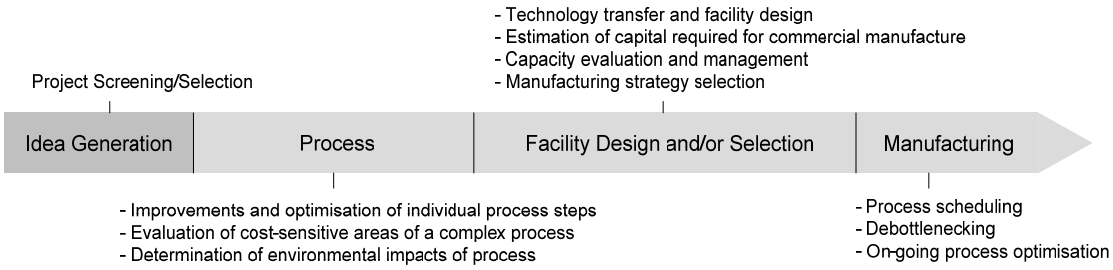


Figure 1.1 Various uses of simulation techniques in stages of drug development

The modelling of biopharmaceutical manufacturing activities however is not a simple task. Saraph (2001) states some key complexities:

- The manufacturing process is a mix of discrete and continuous processes.
- The batch sizes vary from stage to stage.
- Different production stages are physically and temporally separated by intermediate quality control and quality assurance processes.
- Storage capacities at each stage differ.
- Product has limited shelf life at each stage of production and product potency is adversely affected by storage.

- Production capacity differs from stage to stage and so does staffing (in terms of operating shifts and days of the week).
- There are issues of product rejection and process yield.
- There are elaborate controls to ensure required cleanliness, which create further operational constraints as well as other regulatory constraints
- Sharing of common utilities

Due to these complexities and those issues highlighted in section 1.2, decisional support tools play a major part in biopharmaceutical manufacturing. However, despite this, the industry has had little associated simulation work in the past compared to other industries, for example, the chemical or semiconductor industries. As a result, simulation modelling in the biopharmaceutical industry has become very much an ad hoc activity, with very little guidance offered on the approach to creating these decisional support tools.

1.5 Aims and Objectives

The aim of this work is to provide a standard framework to be used as a guideline for the construction of simulation models in biopharmaceutical manufacturing capacity management. The motivation behind this is to create a more cost-effective and efficient way of creating models which meet a certain set of requirement specifications as listed below:

- Intuitive to User
- Relevance and Ease of Data Input/Output
- Short Run Time
- Maximised Reusability and Sustainability
- Minimised Development Time

In order to achieve this, various biopharmaceutical manufacturing case studies will be carried out, addressing both deterministic and stochastic issues involving capacity management.

1.6 Thesis Outline

The structure of this thesis is as follows:

Chapter 2 presents a literature review of different simulation techniques, current platforms available and a critique of their use as decisional support tools. This is followed by a review of current standards available across different industries.

Chapter 3 presents the Standard Framework proposed in this work, presenting its development as part of an evolutionary process. It will highlight the various versions of the standard and the case studies used to test them.

Chapter 4 presents the first of the in-depth case studies used to test the Standard Framework Version 1. The case presented is that of a biosynthetic therapeutic manufacturing facility, looking at scheduling and resource utilisation. Both deterministic and stochastic methods are used to perform process debottlenecking, using Monte Carlo techniques to model inherent risks.

Chapter 5 presents the second in-depth case study used to test the standard Framework Version 2. The case presented is that of a monoclonal antibody production facility, looking at scheduling and resource utilisation. The chapter discusses the scenario analyses used to determine the best strategy to deal with process variability and uncertainty while minimising the downstream cost of column repack and maximising process throughput.

Chapter 6 presents the third and final in-depth case study, using the monoclonal antibody process to look at scheduling in a multiproduct facility to test the Standard Framework Version 2. The scenarios presented here explore multiproduct campaigning, product changeover procedures and operating shifts, looking at the optimal campaigning schedule of multiple products, given a set of constraints, according to three quantitative measures.

Chapter 7 considers the commercialisation of this work and Chapter 8 goes on to discuss the related validation and regulatory issues.

Finally, Chapter 9 concludes the thesis, giving a summary of the work that has been done and discusses possible directions for future work in the area of standardisation in biopharmaceutical manufacturing capacity management modelling.

CHAPTER 2

Literature Review

2.1 Introduction

As discussed in the previous chapter, biopharmaceutical manufacturers are increasingly turning to simulation tools to assist in many design, development and optimisation activities, particularly those related to capacity management, in their endeavour to produce more cost effective and highly productive processes.

Simulation tools due to their graphical representations and inbuilt functionalities (often specific to application area) are popular among simulation analysts and are used at various points throughout the product life cycle. The increased use of these tools in the biopharmaceutical industry has highlighted the need to optimise the actual simulation process, that is, to improve the process by which the models are built. This improvement would mean a quicker model construction time frame, by means of a more efficient use of inbuilt tools and functionality and perhaps a central framework which would provide the foundations for all future models with certain recognised commonalities.

Section 2.2 will review the key aspects of simulation modelling, including model characterisation according to system definition and model dimension. It will then go on to discuss simulation software selection. Section 2.3 will review the current simulation solutions available for both the biopharmaceutical and other industries facing similar problems, analysing the differences and similarities between these solutions. Finally, Section 2.4 will examine the different standardisation approaches available in creating a more uniform approach to the simulation modelling, providing a review of their use in creating useful models.

2.2 Simulation Basics

2.2.1 Systems

This section discusses the basic features of simulation, looking at the different systems and dimensions which can be used to define the different types.

- *Discrete* simulation systems have instantaneously changing variables at separate points in time (Law and Kelton, 1991). That is, the system state variables remain constant and change only at certain points in time called 'events' (Banks, 1998).Examples are queuing systems containing entities, resources, and the queue (e.g. a bank) and manufacturing systems with production parts competing for resources.

- *Continuous* simulation systems involve continuously changing variables with respect to time (Banks, 1998). For example a weather event such as a storm would be a continuous system, or the movement of water through a series of reservoirs and pipes.

2.2.2 Dimensions

When aiming to develop a simulation tool it is also necessary to consider the different dimensions, of which there are three (Law and Kelton 1991):

- *Static versus Dynamic Simulation Models*: Time distinguishes these types of model. Static models represent a particular moment in time or represent systems in which time plays no role. Spreadsheet models created in Excel are an example of this. Dynamic models however represent systems as they evolve over time such as the movement of product through the manufacturing process. A process modelled in Extend would be an example.

- *Deterministic versus Stochastic Simulation Models*: Deterministic models do not contain any random components. The output is determined once the inputs and relationships have been specified. Stochastic models have some random input components and the outputs produced are themselves random.

- *Continuous versus Discrete Simulation Models:* Discrete and continuous simulation models are defined similarly to discrete and continuous systems. However discrete models are not necessarily used to model discrete systems and vice versa. For example, continuous simulations can also be used to simulate systems consisting of discrete entities if the number of entities is large enough so that the movement can be treated as a flow.

2.2.3 Simulation Studies

When presented with a situation deemed suitable for simulation it would be helpful to go through a predefined and tested route for model creation. This route should clearly define the series of steps leading to the implementation of the model. Banks (1998) suggests the steps shown in Figure 2.1.

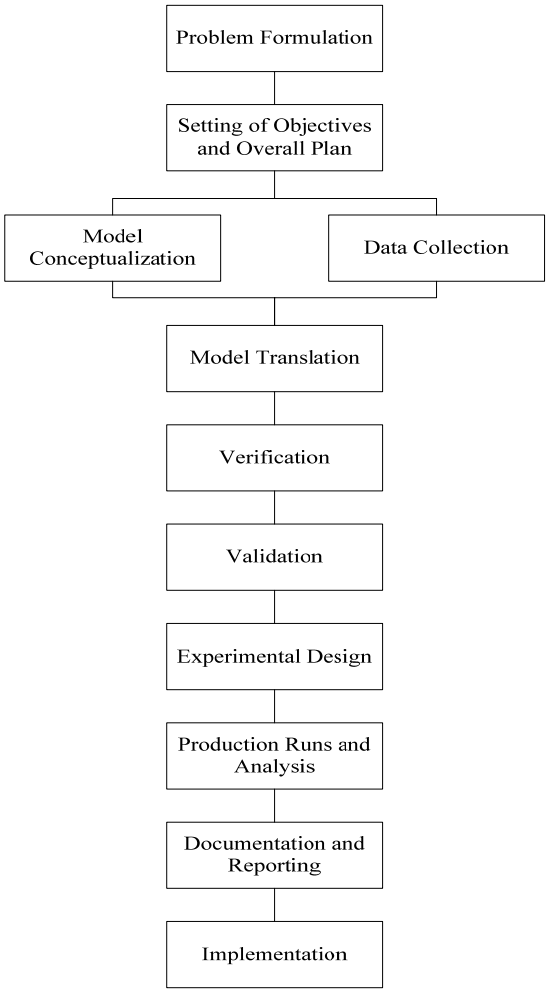


Figure 2.1 Steps in a Simulation Study (Source: Banks, 1998)

Ryan and Heavey (2006) present a slightly different view of the modelling process as shown in Figure 2.2. Applying the ‘‘40–20–40’’ rule they state that, in model development, the modeller’s time should be divided into: 40% to requirements gathering, 20% model translation and 40% experimentation such as validation and verification. Although the main steps are very similar to those presented by Banks (1998), the main difference are the highly iterative nature and the increased emphasis on continuous verification and validation.

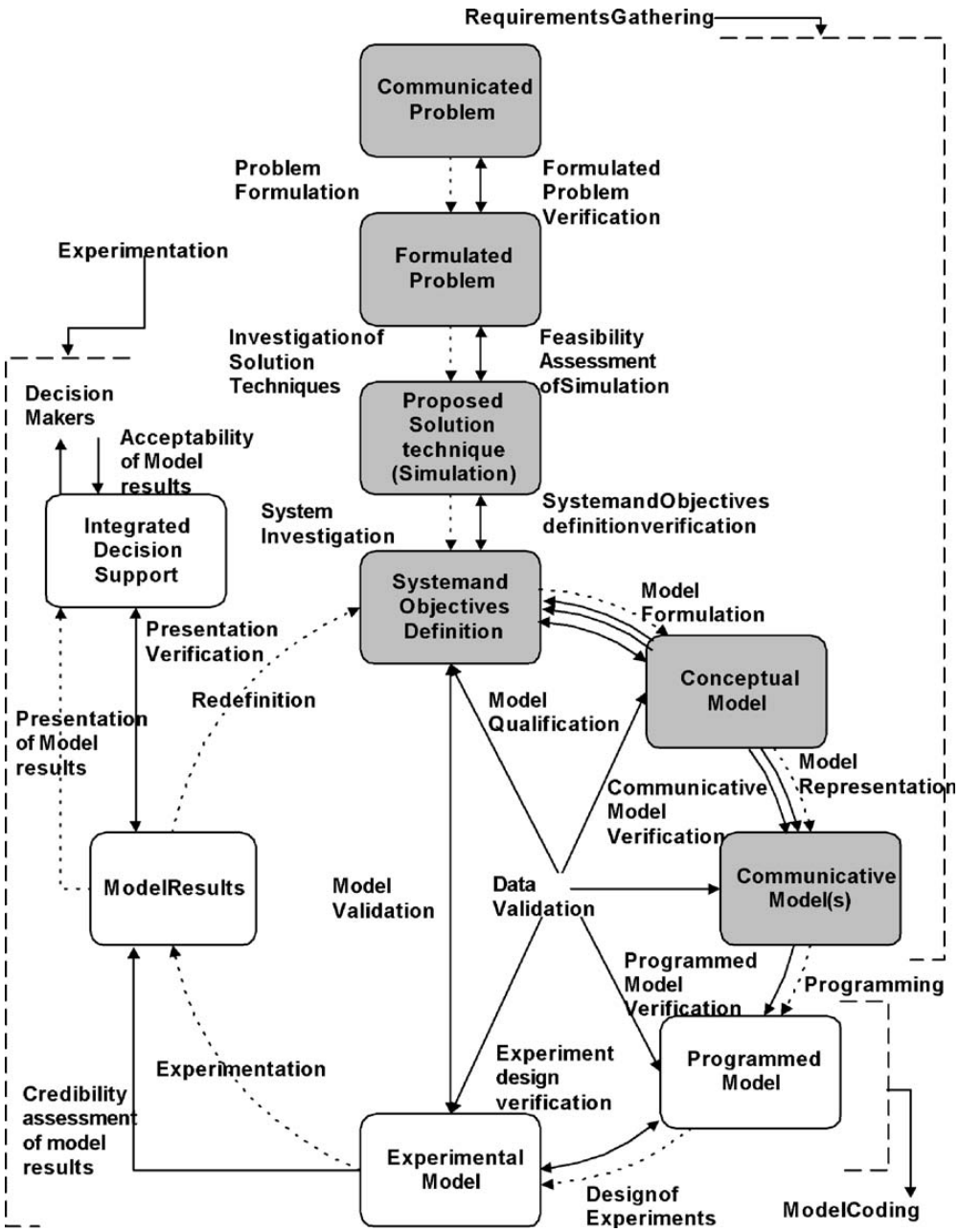


Figure 2.2 The life cycle of a simulation study (Ryan and Heavey, 2006)

2.2.4 Types of Simulation Environment

The term simulation environment is used to describe the platform on which the simulation is carried out, or the type of software used. Figure 2.3 shows the three main types: spreadsheets-based, simulation languages and simulators, which can be further broken down into sub-categories. The following section will discuss the types of simulation environment.

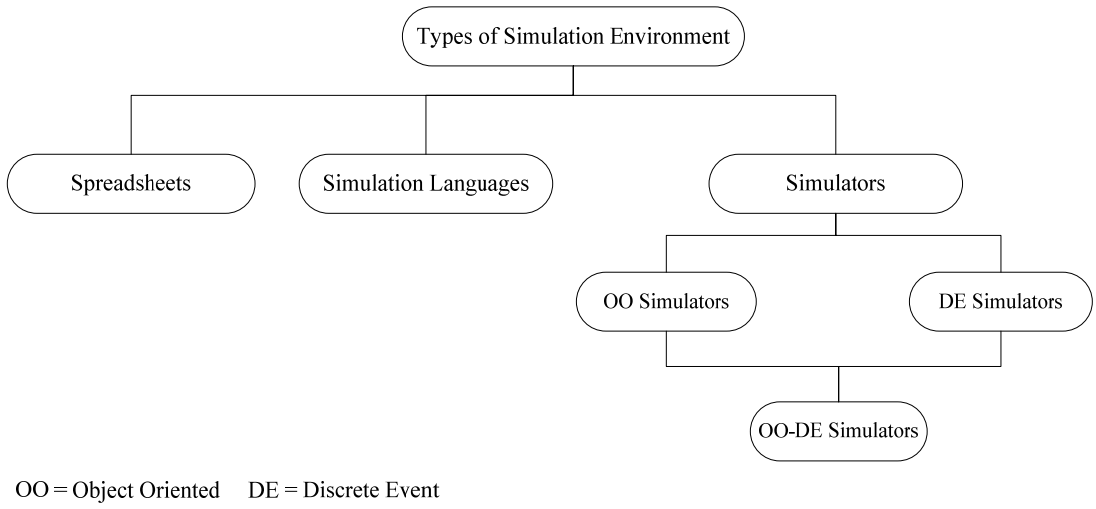


Figure 2.3 Types of Simulation Environment: Spreadsheets, Simulation Languages and Simulators, with the sub-types object-oriented and discrete event simulators.

2.2.5 Spreadsheet-based models

Spreadsheet modelling describes the use of software such as Microsoft Excel to perform comparatively basic system calculations. This type of modelling environment allows for both deterministic and stochastic analysis and optimisation, more so with add-ons such as Crystal Ball (Oracle), @Risk(Palisade) or RiskOptimiser (Palisade). Also, it has the advantage that almost all users will have a spreadsheet tool available to them. However there are limitations to this type of simulation environment. For example, spreadsheets are static in that they cannot model time elements of a system. Thus any dynamic requirements are not met. Furthermore, complex algorithms are difficult to implement, spreadsheets are slower than some alternatives and data storage is limited (Seila, 2001)

2.2.6 Simulation Languages and Simulators

A simulation language uses a computer package that is general in that any system may potentially be modelled on it. A program is written using the language's model constructs which means that extensive knowledge of the language is required.

A simulator however is parameter driven and requires little or no programming (Banks, 1991). Simulators have various types; object-oriented, discrete event, non-object- non-discrete event and object oriented-discrete event. Object oriented simulators model the interacting behaviour of objects over time (Joines, 1998) while discrete event simulators model the state changes which occur at specific points in time.

In recent years there has been a movement towards environments which use simulation languages and a graphical model-building approach, offering a hybrid form of the two simulation techniques. Programme development time is greatly reduced as models are constructed using inbuilt system component libraries. Those simulation packages which do not allow for additional programming have a major drawback in that the models are limited to the system configurations offered by the package. There are however many packages (such as Extend, Imagine That!) which contain their own vast libraries of system components and also allow for programming (using the ModL language) to alter components or to create entirely new ones.

2.2.6.1 Object Oriented Simulation

In this type of simulation the system dynamics are expressed in terms of objects (actors) that exist in parallel and that interact with each other. Every object is represented by: parameters, attributes (also called internal attributes or value attributes) and methods (also called actions or procedure attributes).

2.2.6.2 Discrete Event Simulation

Law and Kelton (1991) describe this as the modelling of a system as it evolves over time by representing the instantaneous change in the state variables at separate points

in time where events will occur. The most common modelling elements in a discrete event simulation system are described as follows

- *Entity*: This can be a dynamic item that arrives to a system and usually exits it. For example, in a model of the downstream processing of an antibody, the entity would be the fermenter harvest stream which would move from unit operation to unit operation eventually leaving via the final step (perhaps a chromatography step depending on the scope of the model).
- *Attribute* (or parameter): A piece of information that describes an entity. Entities may share common attributes, for example the components of a fermenter feed will all share the same fermenter arrival time.
- *Resource*: Resources provide a service to dynamic entities (Banks, 1998) and are required to run the system. Examples include people, machines and buffers.
- *Queue*: If a dynamic entity is denied a resource because it is being used elsewhere then that entity will join a queue for that resource.

2.2.7 Simulation Software Selection

Simulation software must meet certain criteria in order to be of use to the modeller and the 'client' (internal or external). In other words, the software should provide the functionality required to a) create a sufficiently true representation of the real system and b) provide a 'user-friendly' modelling environment.

According to Banks (1998) the following should be taken into consideration when choosing simulation software:

Input

- File Importability - In more complex systems or processes it is useful to be able to import files containing the information rather than re-entering again.
- File Exportability - This feature may be useful when for example graphical representation of the output data is required.
- Syntax - Easily understood, consistent and unambiguous. For example a queue block represents an actual wait in line.

- Interactive Runner Controller (IRC) or Debugger - This assists in finding and correcting errors

Process

- Powerful Constructs – To aid in the representation of the real system.
- Speed - This should remain within reason even for highly complex systems with many entities.
- Random Variable Generator - Usually from a selection of 12 distributions. Law and Kelton (1991) state that statistical capabilities are needed as real world systems show random behaviour, a simulation package should contain a variety of standard distributions to generate random inputs.
- Attributes and Global Variables - Attributes are associated with local entities while global variables are associated with all entities.
- Programming - For a model to truly represent a complex real system a degree of internal programming is required to code the 'complex decision logic' (Gibson). Simply using inbuilt representations will limit the validity of the model.

Output

- Standardised Reports - These can be produced automatically or by request.
- Customised Reports - Tailored to the specific needs of the project.
- Business Graphics - For example bar charts and histograms.
- Database Maintenance.
- Custom Performance Measures.
- Write to a file - It may be useful for the simulator to be able to import a file containing data, events or system variables into a spreadsheet for further analysis or manipulation.

Environmental

- Ease of Use.
- Ease of Learning.
- Quality of Documentation.
- Animation Capability - Elements of a system are represented by icons that alter in some way when there is a change in state in the simulation. According to Law and Kelton (1991) some advantages of having animation are:

- Debugging
- Model validation
- Suggesting improved operational procedures
- Understanding the dynamic nature of the system
- Training personnel

2.3 Current Solutions

2.3.1 Overview

While the focus of this chapter is on biopharmaceutical manufacturing it is useful to consider that industries such as the chemical and food & drink industries face similar problems and therefore adopt simulation techniques in much the same way.
This section reviews the applications of some available commercial simulation packages.

2.3.2 Application of Simulators

ProModel from ProModel Corp (Orem, UT) is a discrete-event (i.e. dynamic) simulation software used for evaluating, planning or designing manufacturing, warehousing and logistics. It is a Windows-based system with a graphical interface and object-oriented modelling constructs that can eliminate the need for programming (Benson, 1997).

The user interface is in 'spreadsheet' format. While complex features of manufacturing systems can be built from pre-existing model elements within the software, programming using languages such as C may be carried out externally and linked to ProModel and accessed any time during runtime for increased flexibility (www.promodel.com).

Companies who have used the software include Pfizer, Johnson & Johnson and Bristol Myers Squibb. A case study released by ProModel Solutions (formed from 2000 merger of ProModel Corp and QuestOne Decision Sciences) describes the use of ProModel by independent consultants to increase the efficiency of a generic

pharmaceutical firm's QC laboratory throughput. A series of 'what if' scenarios were used as a form of sensitivity analysis of the performance of the lab, answering such questions as

- The impact of changing the mix of sample releases on lab efficiency
- Staff shift assignments and team organisation of the operating efficiency
- Identification of production bottlenecks
- Effect of planned and unplanned equipment downtime on lab workflow

(www.promodel.com).

ExtendSim (Imagine That!) is another visual simulation tool which allows model building using pre-built components without the need for programming (Krahl, 2000).
Extend models are built using blocks which describe calculations or process steps. Groups of blocks with similar characteristics can be found in block libraries incorporated in the software. The internal database also allows for a large amount of parameter storage within the software itself and can be accessed by all library blocks.

There are certain features of the ExtendSim software which include reusability of blocks within each model and the ability to save within the system for other models, ability to process large scale system models, simple graphical representation and interactivity with other applications such as Excel. Extend also has an optimisation block which determines the best model configuration. Further flexibility comes from its programmability feature – its built-in C-based language, ModL, can be used to develop new modelling components (Krahl 2000) thus allowing for greater specificity for the user and the project. This integrated programming feature distinguishes it from simulation tools such as ProModel which allow for external programming of model components which are then linked to the model.

An example of the application of ExtendSim is given by Sharda and Bury (2008) where they used the simulation software to determine the reliability of a chemical plant by understanding the key equipment components that contribute towards maximum production losses. According to the Sharda and Bury, Extend was chosen for this application due to its capabilities in modelling both discrete and continuous process elements simultaneously and also it's hierarchical structure, allowing for

intuitive model. In order to determine plant reliability they ran a series of scenarios based analysis, using the Extend database to store and export data for analysis. As a result they were able to determine areas of concern with regard to component efficiency.

Lim et al (2004) describe the use of Extend to model and compare perfusion-based and fed-batch based processes in the production of monoclonal antibodies, looking at features such as resource management, mass balance analysis, in-process testing and costing. Lim et al (2004) use the database SDI tool to transfer data via the interface with Excel. This is a complex method of data transfer and it requires an expensive plug-in for Excel. However for the more complex models with a great number of Excel tables which need to be linked it is the better choice as it allows for automatic importation of all similarly structured tables and ease of use when relating the data or attributes to items. With regards to calculations, instead of carrying these out in Excel and then linking the results of those calculations to the Extend model via tables, calculations are carried out in Extend. Lim (2004) comments on the limitations of this approach.For a complex model calculations done in Extend slow the simulation runs down significantly. In fact a run of this particular model took 10 minutes on an average processor rather than a few seconds. Calculations done in Extend also often cause the software to crash which is highly inconvenient. However, Lim et al (2004) also comment on the advantages of using the Extend simulation tool stating that the graphical representation creates an intuitive simulation environment while the animation during model runs enables the user to view events as they occur and allows for debugging.

Rajapakse et al (2004) use Extend to map the development, manufacture and testing stages of the three clinical trial phases along with the proceeding 20 years in market for 6 drugs. The fundamental aim of the model is to calculate the estimated number of drug successes and NPV based on development times, capital and human resource constraints and taking into account the implications of contract manufacturing. Here, Rajapakse et al (2004) introduce stochastic modelling, adding probability distributions to uncertain input parameters, to perform sensitivity analyses. No limitations of the software tool are stated.

Simul8™ (Simul8 Corporation) is a discrete event simulator which interfaces to common external programs such as Excel or Visio. It is targeted at many industries including manufacturing and capacity planning. A Simul8 simulation involves processing work items. These enter the system via work entry points, pass through work centres, may temporarily reside in storage areas and leave via work exit points (Hauge and Paige, 2001). Hauge and Paige (2001) state that when creating a model using the Simul8 platform, the starting point is to add a few items onto the workspace and connect them, creating the basic logic of the simulation.Further details can then be specified via dialogue boxes, including changing the names of the objects. This functionality is very similar to other discrete event simulators. SIMUL8 can also be used to conduct extensive trial runs. When a trial is complete the software automatically prepares summary reports of the results. Specific areas can then be picked from the summary and further analysed by the various statistical tools and diagrams made available.

The Simul8 simulation tool allows analysis of manufacturing processes without the requirement of programming knowledge. However, importantly, its simplicity creates a utility threshold beyond which more complex workbench tools such as Extend would serve a company better in seeking answers to more in-depth design and manufacturing questions.

2.3.3 Application of Spreadsheet Based Models

Superpro Designer (IntelligenInc) is a static modelling tool which is spreadsheet-based with a graphical interface.

Gosling (2003) describes the use of Intelligen'sSuperPro Designer to construct a flowsheet for the fermentation train of a biopharmaceutical manufacturing process. The resulting flowsheet contains fermenter, storage and media prep vessels with intermediate pumps and also all input and output streams. SuperPro Designer's use of inbuilt bioprocess icons allows for user friendly flowsheet diagrams for simpler models however for more complex systems the flowsheets become far too busy.

Gosling then describes the use of Intelligen'sSchedulePro to generate equipment occupancy profiles for the overall production schedule based on the process and utility information provided by SuperPro Designer.

Petrides (2002) describes the use of SuperPro Designer for simulating the production of an intermediate pharmaceutical compound, looking at the process scheduling, resource tracking, capacity utilisation and economic evaluation of the process using tools provided by the simulation package. One important functionality of SuperPro highlighted here is ability to perform equipment sizing calculations based on material balances and to calculate batch cycle time by estimating the cycle-time of scale-dependent unit operations. Furthermore Petrides (2002) also discusses the tracking of volumetric utilisation of all vessels throughout the batch cycle - this differs from packages such as Extend which calculate utilisation based on time-in-operation during simulation runs.

Biwer et al (2005) describe the use of SuperPro in the uncertainty analysis of penicillin V production, usingthe platform to provide the material balance and key economic parameters. The output was then exported to Excel in order to run Monte Carlo simulations using Crystal Ball. They state since most computations in SuperPro can also be done in spreadsheet calculations, thus transfer from the simulator platform to Excel is made possible, however they also state that it is the most time consuming part and has a certain risk of transcription errors. This means that validation is necessary between the SuperPro outputs and the constructed Excel spreadsheet.

Jully et al (2004) further describe the use of SuperPro Designer to model, optimise and debottleneck a pharmaceutical production process. The experiment largely focuses on the ability to manipulate process parameters such as cooling time and filling rate by replacing actual unit procedures. This highlights a fundamental difference between Superpro and again, Extend. Superpro's in-built unit procedures which represent activities such as fermentation, blending and so on are provided with dialog boxes in which various parameters and variables can be entered. For example material mass balances, temperature, rates etc. In Extend however activities can be represented by one or a series of blocks which do not contain such integrated

attributes per se. Instead combinations of blocks and data inputs allow for activity modelling and other functions such as simple mass balancing.

BioSolve (Biopharm Services UK) is an Excel based bioprocess modelling tool designed for operating costs analysis such as cost of goods, utility sizing and waste generation.Due to relatively new market age of the software there is no literature published as yet which provides an unbiased analysis of the tool (all publications are in collaboration with the vendor) however some conclusions can be reached in review of the software. Firstly, due to the spreadsheet driven nature of the tool, models can be generated relatively quickly when compared to simulators such as Extend. Secondly, what sets the BioSolve software apart from some other spreadsheet based tools such as SuperPro is that the cost and process data is provided and updated by the vendor company. However since the models are static theydo not capture the dynamic nature of the systems they are used to model, such as the complexities of scheduling.

2.3.4 Optimisation Models

An alternative to simulation is the use of optimisation models which primarily rely on mathematical algorithms.

An example of an optimisation approach is given by Lakhdar (2005) where mixed integer linear programming is used on the GAMS platform to determine optimal scheduling of a multi-product facility given variability in parameters and constraints, as defined by the modeller. Although the paper concludes that the model was successfully used to deliver an optimal schedule (with operating profit being the objective function) it is believed that this approach is not as appropriate for all simulation scenarios for the following reasons. Firstly, the heavy reliance on mathematical algorithms requires the problem to be modelled in a very rigid structure. Secondly the question of capacity management and optimal scheduling relies on the interaction of a series of events and constraints which can be better represented using fully dynamic object oriented or discrete event simulators.

Miller et al (2010) describe the use of a biologics facility model built using the VirtECS version 7.1 software system (Advanced Process Combinatorics, Inc., West Lafayette, IN) in order to generate an automated production schedule based on the variability and constraints inherent to the process. The software used consists of a core mathematical programming solver and a customized outer layer that is specifically tailored to address biologics process behaviour. The core solver, which solves a mixed integer linear program (MILP), is used by the outer algorithm which is required to address stochastic variability, through a sampling of distributions, of uncertain parameters such as titre. Miller et al (2010) compare the use of mathematical model to the use of discrete event simulators, stating that the former approach offers two main advantages to the DES method. Firstly, that the level of complexity which arises with simulators is great, due to the high number of equipment and tasks. Secondly, that due to the fact that discrete event simulators only allow events to be processed in chronological order, no events earlier than an occurred event can be considered. They argue that with the mathematical approach considering different demands is a simpler operation due to this functionality, as the solution strategy can range over the timeline moving back and forth until a complete solution is generated.

2.4 Standardisation Approaches

2.4.1 Motivation

The simulation tools described in this review are highly capable of modelling complex systems and have been used across different industries for various purposes.

However many companies and industries in general have recognised the need for a more standardised approach. For example, imagine a large pharmaceutical company. whose manufacturing facilities in America, Europe and Asia each require high degrees of simulation whether for new projects or existing manufacturing systems. Over time this company has accumulated a large number of models each specific to the individual system and problem scope built entirely from scratch each time, thus proving costly in terms of time and labour intensity. With a standard approach in

place that takes advantage of system commonalities, the development time could be significantly reduced allowing for a framework initially applied in a US facility model to be applied in the UK.

A possible solution this problem is to perform an extensive study, identifying areas of commonality between a vast array of models previously built by UCL and other organisations. This will allow for the construction of a model infrastructure, a generic model template which can be used and built upon for almost any aspect of a biopharmaceutical manufacturing activities, thus reducing the cost and time associated with building each new model on an ad hoc basis.

In order to create such a framework it is first necessary to understand the current standardisation approaches available.

This section reviews the current standardisation approaches available, discussing their potential application within the context of the work in this thesis.

2.4.1.1 Manufacturing Simulation and Visualisation Program

The Manufacturing Simulation and Visualization Program at the National Institute of Standards and Technology (NIST) is a US institution developing a framework for manufacturing simulation data standards. They propose that the Banks approach to modelling described in section 2.2.3 leaves too much to the skill of the individual analyst and allows little opportunity for the analyst to build upon the work of others since each simulation is built as a custom solution to a uniquely defined problem (McLean and Leong, 2002). Instead a modularisation approach is proposed whereby re-usable model building blocks are created which can be used for all models reducing the duplication of simulation work. Furthermore an interface for transferring data between simulation and other manufacturing applications is proposed to accelerate the simulation process. A standardisation in language would perhaps be a requisite to the modularisation approach as the standard blocks would only be re-usable if compatible with every application used by simulation analysts.

McLean and Leong (2002) suggest a hierarchical classification approach in looking at a modelling problem, with the major aspects being those described below.

The Industrial Market Sector

This defines the manufacturing industry and describes the end product.

Hierarchical Level of the Manufacturing Organisations System or Process

A meta-hierarchical approach is proposed which attempts to generalise the modelling area. McLean and Leong (2002) justify the partitioning of the meta-hierarchy with the significant differences in the models and data required to simulate each level, however realise that one model may encompass more than one of the levels. The simulation meta-hierarchy from highest to lowest is:

- Economy
- Market – corresponding to the individual sectors, group of sectors, etc. May be used for forecasting demand, prices etc.
- Supply chain
- Enterprise – here this defines the boundaries of the company. Large pharmaceutical companies for example have many facilities many of which are integrated in one supply chain
- Facility – the modelling of departments, activities, equipment at one facility
- Department – engineering, finance etc
- Line, area, or cell – grouping of stations and/or equipment for manufacturing a product
- Station – place where work is performed by person or robot.
- Equipment
- Device – separable elements of equipment e.g. sensors, membranes
- Process – the physical manufacturing process is the lowest level of the hierarchy

The Simulation Case Study

This describes the question that the model would be answering. The following case studies proposed by McLean and Leong (2002) are some areas which a model would be used in:

- Market forecast
- Logistics network
- Site selection
- Work force
- Product mix
- Capacity Analysis
- Tolerance analysis
- Ergonomic analysis
- Tooling

- Business process
- Scheduling
- Plant layout
- Capital equipment
- Line balancing
- Cost estimation
- Process validation
- Process capability
- Inventory
- Material handling
- Maintenance

Model Elements, Input, and Output Data

The detail level of a model and therefore the data required to construct it greatly depends of the case study and the hierarchical level of the organisation or process. The data formats developed by NIST have been divided into:

- General and miscellaneous, e.g. units of measurements, probability distributions
- Organizational structures
- Product and process specifications
- Production operations
- Resource definitions
- Layout

2.4.1.2 Process Specification Language (PSL)

The use of computer software is becoming increasingly commonplace in manufacturing operations and is used in a vast number of areas. As demand of applications has multiplied the information has become more complex with many manufacturing engineering and business software applications using process information, including manufacturing simulation, production scheduling, manufacturing process planning, workflow, business process reengineering, product realisation process modelling, and project management. Each of these applications uses process information in a different way, and therefore their representations of process information are also different. Often they associate different meanings with the terms representing the information they are exchanging (Schlenoff et al, 2000) thus there has been recognised a need for a standard neutral translator to allow these applications to interoperate taking into account their semantics (meanings of their terminologies) and their syntax.

PSL is an undertaking by the National Institute of Standards and Technology (NIST), the aim of the project being to create a process specification language which

will be common to all manufacturing applications and robust enough to represent the required process information. PSL should act as a medium, facilitating communication between different organisations or different branches or sites of the same organisation; currently these organisations have to use the same software so as to be able to communicate process representations between them, a problematic solution if one of the organisations is not familiar with the package. The concept behind PSL is that the organisations will be able to use two different packages which are 'PSL compliant' and be able to transfer files with common language and representation components with ease facilitated by using the PSL ontology, that is, the set of terminology along with their meanings existing in the PSL lexicon.

Schlenoff (2000) defines a language as a set of symbols and a grammar (a specification of how these symbols can be combined to make well-formed formulas). The PSL lexicon consists of logical symbols (such as Boolean connectives and quantifiers) and non-logical symbols. For PSL, the non-logical part of the lexicon consists of expressions (constants, function symbols, and predicates) chosen to represent the basic concepts in the PSL ontology.

The PSL project findings offer a gathering of requirements necessary for modelling a manufacturing process. According to Knutilla (1998) there are 4 major categories of requirements:

- **Core**: Most basic and intrinsic to all processes. Provide basis for representing only simplest of processes e.g. resource, task
- **Outer Core**: Pervasive but not critical. These requirements describe processes such as resource grouping and task alternatives.
- **Extensions**: Grouping of related requirements which together give added functionality. Six extensions are Admin/Business, Planning/Scheduling/Quality/ Analysis, Real-Time/Dynamic, Process Intent, Aggregate Resources/Processes, and Stochastics/Statistics.
- **Application Specific**: Only relevant within specific applications e.g., dynamic rescheduling for the production scheduling environment

Although PSL is targeted at manufacturers who need to exchange process information, ultimately it is the software vendors who would need to develop and incorporate the PSL translators into the simulation tools (www.mel.nist.gov/psl). However the concept of PSL is still applicable to the modelling process; a standard ontology for a standard framework can aid in its consistency and its intuitiveness to the modeller and end user.

2.4.1.3 Unified Modelling Language (UML)

The Unified Modelling Language (UML) is an object-oriented specification language that began in the mid 1990s as a solution to the vast number of languages which individually failed to completely satisfy the requirements of object oriented methods. It is a standard language for specifying, visualising, constructing and documenting the artefacts of software design (Siau and Cao, 2001) by using a standardised set of symbols and ways of arranging them. Its use in systems engineering is of particular interest here as it focuses on using a set of diagrams to represent a system from different viewpoints (Ramos 2003), implicating a potential as a tool during the design phase of the modelling activity.

Ramos (2003) describes the use of UML as a structural modelling tool, used to identify the key operational components of a cellular manufacturing system for the construction of a system template. Split into two phases, the initial identification of operating parameters and their relations was done through the use of UML class diagrams. The second phase, the definition of the simulation modelling constructs was done through the use of the UML components diagrams. Therefore the standard was used both in the system definition and the model design stages, allowing for model component selection i.e. definition of the scope of the template, allowing for the construction of a template that was reusable, extendable and intuitive to the user (Ramos, 2003).

Glinz (2000) describes a case study looking at the application of UML to the Teleservices and Remote Medical Care System (TRMCS) which has a certain set of systems requirements. The TRMCS provides medical assistance to at home or mobile patients and Figure 2.4 shows the use case diagram for the system:

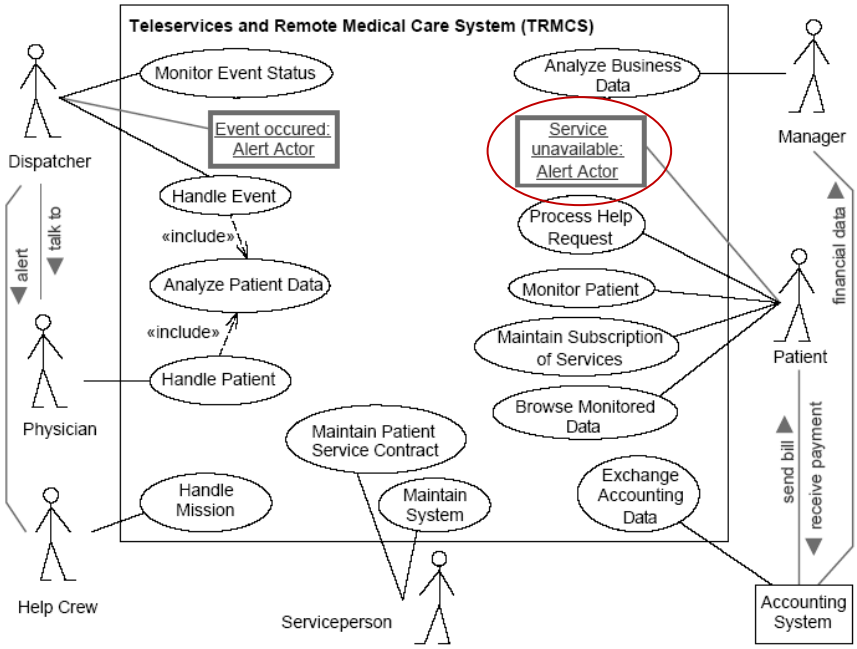


Figure 2.4 Use Case diagram of the TRMCS

In reality the TRMCS should warn a mobile patient when a service becomes unavailable because she or he is moving out of range of the mobile communication network. So an active model element, for example, an active object is needed which is able to initiate communication between the system and an actor (circled area in Figure 2.4). However, such an element cannot be modelled in UML: a use case by definition describes a sequence of actor stimuli and system responses *that is initiated by an actor* (OMG, 1999). Active objects are not allowed in UML use case diagrams. Therefore the first deficiency of UML is that a use case model cannot specify the interaction requirements where the system initiates an interaction between the system and an external actor (Glinz, 2000). Another problem highlighted is that UML cannot model rich system context i.e. because UML does not allow associations between actors it is not possible to model the important necessary external interactions such as a Dispatcher communicating with a Physician. Put into a pharmaceutical context this would mean that for example the communication between departments could not be modelled in a SOP workflow model. However, UML is not a software process.

Furthermore Seila (2005) argues that the many graphical representations are quite abstract and none can be used directly to model the operational dynamics of the system from a discrete event simulation perspective.

Ryan and Heavey (2006) state that while UML activity diagrams are capable of representing workflow and dataflow within a discrete process they do not visually account for detailed interactions or the complex use of resources within a detailed simulation model.

A conclusion that can be made from this review is that the UML standard, specifically the class diagrams, can be used in system definition phase of the model construction process, helping to clearly define the model scope. Barjis and Shishkov (2001) also propose the use of the UML activity diagrams as a pre-simulation technique. However the standard is limited in its ability to fully define the actual model construct.

2.4.1.4 Integrate DEFinition, IDEF0

The purpose of this standard is to aid in the construction of a system model that consists of a series of hierarchical diagrams with the primary modelling components being functions (boxes) and data objects which link the functions (arrows). While the detailed concept of the IDEF0 standard cannot be applied to all object oriented DE simulation modelling, there are some useful guidelines provided which can be used to construct a more standardised model. The main rules have been listed:

Top-level Context diagram: The standard states that the top level contains a single box to represent the subject of the model. The standard also states that inputs and outputs to this block should represent the broader picture, in order to represent the context of the system being modelled.

Child Diagram: This is how the standard refers to the sub-levels i.e. when a function is decomposed into its sub-functions. So the workspace visible below the top-level diagram will be the child diagram of the top level parent diagram. Each level can be both a parent diagram and a child diagram (except the top level which can only be a parent).

Numbering blocks at each level: A node number is based on the position of a box in the model hierarchy.

Number of child boxes per child diagram: The standard states that there should be 3-6 child boxes on each child diagram. In other words, 3-6 sub-functions on the workspace for each decomposed function.
Connections: The standard states that boxes be connected by solid arrows

Al-Ahmari and Ridgway (1999) describe the use of IDEF0 to create a modelling method for manufacturing systems analysis and design, applying the standard in the creation of a static model during the design phase of the modelling process and also the simulation construct. They specifically use the IDEF0 standard to analysis different sub-activities and their inputs, outputs, controls and mechanisms. The IDEF0 model of every activity is decomposed into more detailed diagrams until the activity is described in the necessary level of detail (as defined by the scope). The resulting IDEF0 model is a hierarchy of diagrams derived from the decomposition of the activity. The conclusions reached with regard to this use of the modelling method are that 1) the conceptual views define the system functions and their internal activities which outlines decision problems and general configuration difficulties and 2) the functional structure summarises system activities and sub-activities illustrating their relationships, indicating any misinterpretations and inconsistencies in the model upon model validation (Al-Ahmari and Ridgway, 1999).

Ryan and Heavey (2006) discuss the use of the IDEF0 standard as a descriptive method in discrete event simulation modelling. They state that although the standard allows for the visual modelling of the decision and activities of a system, it lacks the ability to model some aspects of a complex discrete event system such as workflow or control flow. They also state that the method doesn't allow graphical representation of the division of a system into multiple processes.

2.4.1.5 Industry Standard Architecture, ISA-88

The purpose of this standard is to provide a standard terminology and a consistent set of concepts and models for batch manufacturing facilities in order to improve

communications between all parties involved (ISA 1995). ISA-88 describes a process or a facility in a physical model, defines what needs to be done through batch recipes, implements recipes using equipment logic and coordinates these steps in a reusable way.
The standard first describes the hierarchical subdivisions of a batch process. This hierarchical breakdown divides the process into stages, operations and actions and the overall process model describes the process requirements.

It then goes on to define the physical model which describes the physical assets of an enterprise

- **Unit**: One or more major processing activities take place in a unit e.g. cell disruption. Also it is presumed that a unit will operate on only one batch at any one time.
- **Equipment Module**: This may be part of a unit or a stand-alone equipment grouping within a process cell. An equipment model can carry out a finite number of minor processing activities such as dosing and can be shared between units.
- **Control Module**: Typically a collection of sensors, actuators and other control modules.

So if taking the production of a monoclonal antibody as an example this ISA-88 terminology can be used in the following way:
- Unit: e.g. Fermenter
- Equipment Modules: agitation system, probes
- Process Cell: Process chain consisting of fermenter, centrifuge, chromatography columns etc

ISA-88 also describes the way in which resources should be allocated and arbitrated. It states that resources are assigned to a batch or a unit as they are needed to complete or to continue required processing with allocation controlling these assignments. When more than one candidate for allocation exists, a selection algorithm such as "select lowest duty time" might be used as a basis for choosing the resource. When more than one request for a single resource is made, arbitration is

needed to determine which requester will be granted the resource. An algorithm such as "first come/first served" might be used as a basis for arbitration (ISA 1995). This is a functionality which can be found in Extend; set priority blocks and decisions blocks decide upon resource usage and item process path.

Pandiana (2002) describes the use of ISA-88 in the hierarchical decomposition of the process model for a margarine plant. First the IDEF0 process definition is used to decompose the production process into subsequent sub-process stages as shown in Figure 2.5.

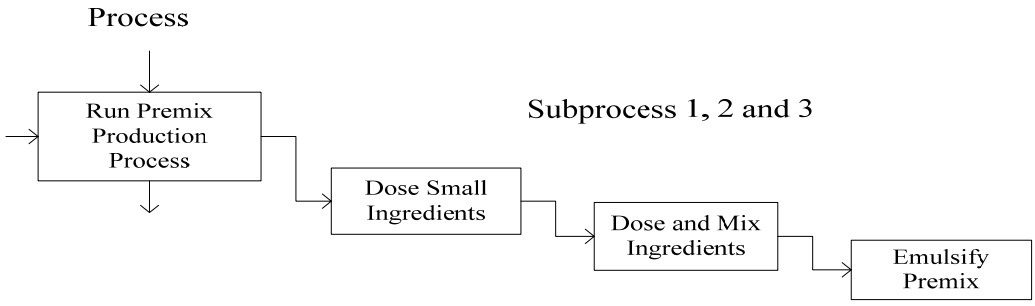


Figure 2.5 IDEF0 Decomposition of Overall Process for Margarine Production (Source: Pandiana, 2002)

Following on from the IDEF0 process model a tree structure depicting the process steps can be drawn. The activities, according to the ISA-88 terminology can be considered to be operations performed by a unit. The ISA-88 units by the standards definition should be able to contain material. For the determination of units therefore it is useless to decompose activities performed by equipment which do not contain material such as control activities. According to Pandiana (2002) this method of decomposition and unit determination allows for more efficient measurement of performance indicators by avoiding too much detail and concentrating on the areas of the production process which should be checked.

2.4.1.6 Industry Standard Architecture, ISA-95

The ISA-95 standard is in large part an extension of the concepts and terminology defined under ISA-88. While ISA-88 defines the models and terminology on the factory floor, ISA-95 is a standard between the factory floor and the enterprise It therefore addresses the interface between Business Planning and Logistics level (level 4) and the Manufacturing Operations and Control level (level 3) of the functional hierarchy defined within the standard (ISA 2000).
There are currently three parts to the standard:

- Part 1: Models and Terminology
- Part 2: Object Model Attributes
- Part 3: Manufacturing Operations and Control

Part 1 is of particular interest here as it offers a set of terms and models which can be applied to the pre-model construct and model construct stages. Figure 2.6 shows the ISA-95 equipment hierarchy which defines the different organisation levels leading down to the process and equipment. Again, ISA-95 deals with level 4 while ISA-88 deals with level 3.

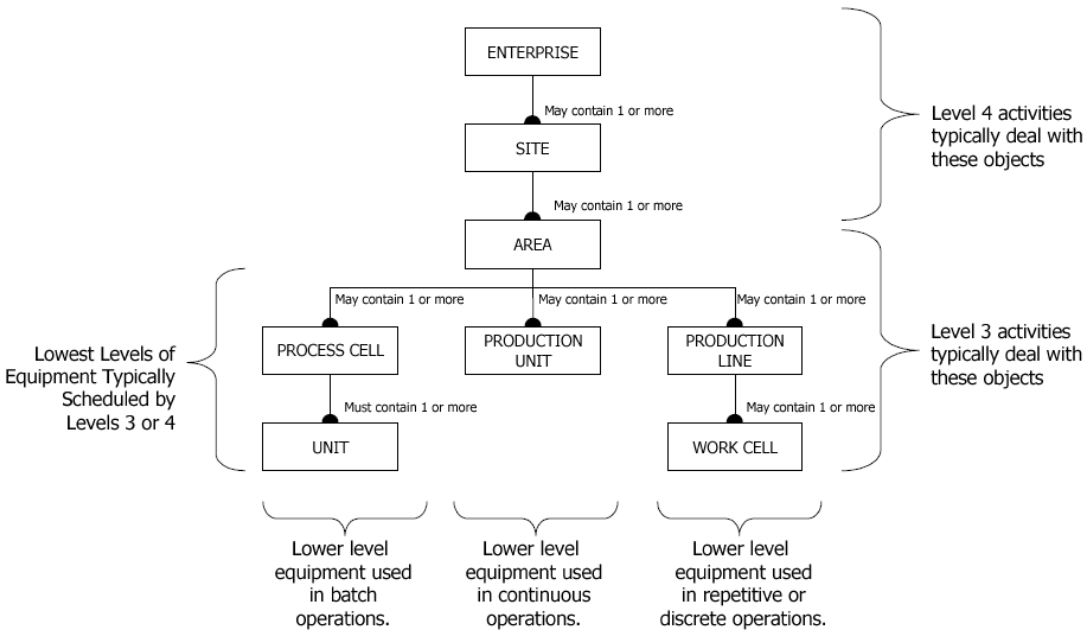


Figure 2.6 ISA-95 Equipment Hierarchy

Walker (http://www.idef.com/idef0.html) briefly describes how Dupont Engineering used the ISA-95 operations diagrams to map out a project scope, decomposing this further, constructing a spreadsheet defining manufacturing operations and their breakdown into processes, functions and tasks. Walker goes on to describe how the standard was used to measure the business value of the project and to analyse supplier capabilities using part 2 and 3.

2.4.1.7 The Business Process Modelling Notation

The Business Process Modelling Notation (BPMN) is a standard graphical notation. Through a limited set of shapes, associated semantics and connectors, it is a method of standardising the way in which business processes are modelled. The following describes the main features of the BPMN standard.

Events

There are three types of event:

- Start : An event which will trigger the execution of a business process model instance
- Intermediate: This event indicates when an event might happen during the execution of an instance
- End: Indicates where and how a process flow ends.

Activities

The standard defines an activity as a unit of work to be performed. It might be a task, a process or a sub-process. They are represented by a rectangular shape. Markers are defined as well to specify additional semantics such as loops.
The BPMN specifications give the following definitions

- A **Task** is an atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail. Generally, an end-user and/or an application are used to perform the Task when it is executed.
- A **Sub-Process** is Process that is included within another Process. The Sub-Process can be in a collapsed view that hides its details. A Sub-Process can be in

an expanded view that shows its details within the view of the Process in which it is contained.

Gateways

The BPMN defines gateways as decision points used to constrain the execution flow, fork an execution point into several or merge several into one.

Workflow Patterns

BPMN defines this as the different situations where a specific state of process is used to affect its execution. The following describes the different basic control flows:

- **Sequence:** The sequencing of activities in a series formation. The majority of activities in biopharmaceutical process chains will be of this type.
- **Parallel Split:** The parallel formation of activities. In biopharmaceutical activities, this parallel formation takes place where there a single process function is carried out by more than one equipment,
- **Synchronisation:** used to merge two parallel flows**.** This method is used when two activities must not only begin together but end together
- **Exclusive Choice:** where an exclusive gateway is used to create a decision point or the routing of flow based on a decision.

Onggo (2009) describes the use of the BPMN standard in a unified conceptual model representation of a healthcare system with specific application to the problem formulation stage i.e. the pre construct modelling phase. The study concludes that the standard has the ability to model complex business processes, it is a process-centric approach is intuitive and easy to understand, it is scalable by supporting hierarchical decomposition of the processes and allows plausibility checks to be carried out on the diagrams thus allowing verification of problem formation.

2.5 Conclusions

In conclusion, there are a number of different modelling methods, namely simulators, spreadsheet-based tools and optimisation methods. The application of these in the bioprocess industry has increased over the last few decades and a number of case studies carried out. Due to the rigidity in structure of the mathematical approaches and the limited dynamicity of the spreadsheet-based tools, discrete event simulators are the preferred platforms for creating dynamic models capable of representing uncertainties and constraints along with a more user friendly modelling environment.

Furthermore, it seems that there are various existing standards to aid in the simulation process. However, as the next chapter will discuss more fully, only a small proportion of their components can be applied to the standard being proposed in this thesis. The most significant factor being that while these standards may be suitable for other industries, they do not take into account the complexities of biopharmaceutical manufacturing which were highlighted in Section 1.3.

CHAPTER 3

Development of the Standard Framework: An Evolutionary Process

3.1 Introduction

As indicated in Chapters 1 and 2, there is a growing need in the biotechnology industry for simulation tools to manage and improve capacity utilisation in manufacturing facilities across a range of areas such as bulk product manufacture, fill-finish operations and QC lab operations. This chapter presents the design of a standard framework for building such tools so as to facilitate rapid development of models that are easy to understand, re-use and extend by other model developers.

The latest generation of DES packages such as Extend, Simul8, ProModel and Simple++ have graphical user interfaces (GUIs) and in built block structures which means that more people are able to develop models with little or no programming skills required. The nature of the new packages also means that customers and users of the models are more easily able to understand them, increasing their value as decision support tools. It is important to note however that whilst the GUI and in-built functions are useful, the benefits do not occur simply by using these alone.

They instead have to be designed into the models using these new characteristics and whilst model development is a much simpler task than prior to the existence of new generation DES packages, the task of developing a useful and appropriate model for the intended use remains far trickier than one would like. Therefore a structured

approach to DES model building is required. Having to create new models from scratch results in a portfolio of models varying vastly in design and yet representing systems which may have commonalities which would allow for common model elements and/or structure. Therefore a structured approach or framework for DES modelling supported by guidelines, standards and software tools and blocks is required which will lead to more efficient and rapid development of well constructed models which take into account these commonalities. These models must also be understood, updated, re-used and inherited by others (Oscarsson and Moris 2002). In order to form the structured approach an understanding is required of the objectives of any biopharmaceutical model.

This chapter will specifically focus on the DES software Extend (ImagineThat!, San Hose, California). Extend allows the use of inbuilt blocks to represent any dynamic, real system with the added availability of code manipulation for more specific and tailored modelling using the ModL language.

The following sections will discuss the general features of a biopharmaceutical process, an understanding of which will allow definition of the scope of the proposed framework designed to aid in meeting a defined set of requirement specifications.
This work will describe the development of the standard framework as part of an evolutionary process. Sections 3.2 to 3.4 will describe the features of the standard framework which are the domain description, model requirement specifications and the model and system elements which contribute in meeting them. Section 3.5 will describe the evolution of the standard framework.

3.2 Domain Description

DES models are used to create representations of ‘systems’. It is important to note that for the purposes of this work, system refers to the specific dynamic domain which any particular model will represent. Thus a clear understanding of this domain is required in order to define a comprehensive and relevant set of standards and guidelines.

McLean and Leong (2002) suggest a hierarchical classification approach in looking at a modelling problem, with four major aspects: The Industrial Market Sector, the Hierarchical Level of Manufacturing Organisations, Systems or Process, the Model Elements, Input and Output Data and the Simulation Case Study. Application of Mclean and Leong approach has been proposed for the semiconductor industry by Li et al (2005), extending their definition to cover Supply Chain.

While the above hierarchical approach covers the basic stages of tackling a modelling problem, consideration of actual real-world modelling projects suggests that it is by no means comprehensive, giving little guidance as to the link between the system and its representative model or the level of detail required. Thus a new hierarchical approach is proposed based on the same concepts however attempting to distinguish between the hierarchical levels inherent to the real problem and the modelling solution. Importantly it shows that there are further levels of an organisations hierarchy which must be taken into account. According to the standard ISA-88 an organisation can be broken down into seven levels: Enterprise, Site, Area, Process Cell, Unit, Equipment Module and Control Module. Adapting this definition to the biopharmaceutical environment gives the following levels in a pharmaceutical organisational hierarchy: Supply Chain, Enterprise, Product, Site, Department, Process and Equipment.

Furthermore in dynamic systems modelling, at each level any model will represent the flow of object(s) (items, resources etc). Depending on the intended use and the level of detail required to satisfactorily answer the question being asked, this entity flow will only require modelling of a maximum of 4 levels in the hierarchy. The number of levels will decrease higher up. For instance, if the ISA-88 method of decomposition is to be followed then every model should be ultimately decomposed to the unit operations or in this case the equipment level. However it is important to note that a simulation problem classified at the top of the organisation hierarchy does not necessarily model all those levels below it. For example a model looking at supply chain will probably never require the level of detail which would entail modelling the process and equipment. A model looking at a site may require modelling of the building and/or the processes however, again, not the equipment.

The level of organisational hierarchy along with the type of system and the scope of the problem ultimately help define the level of detail and specifications of the model needed in order to give a true representation of the real system.

As Figure 3.1 shows, in order to classify the model and therefore define the type of model required (simulation, spread-sheet based or optimisation and platform to be used) it is first necessary to classify the system i.e. manufacturing, transactional flow, QC laboratory and so on. The system classification and the scope, as defined by the client or model user, will define the data requirements. For example, a query into the number of freeze dryers needed to reduce campaign cycle time by 20% in a fill-finish facility will require data such as cycle times, labour requirements and rules regarding dryer usage. Once clearly defined, the level of detail in the model is intrinsically decided by the scope and data requirements.

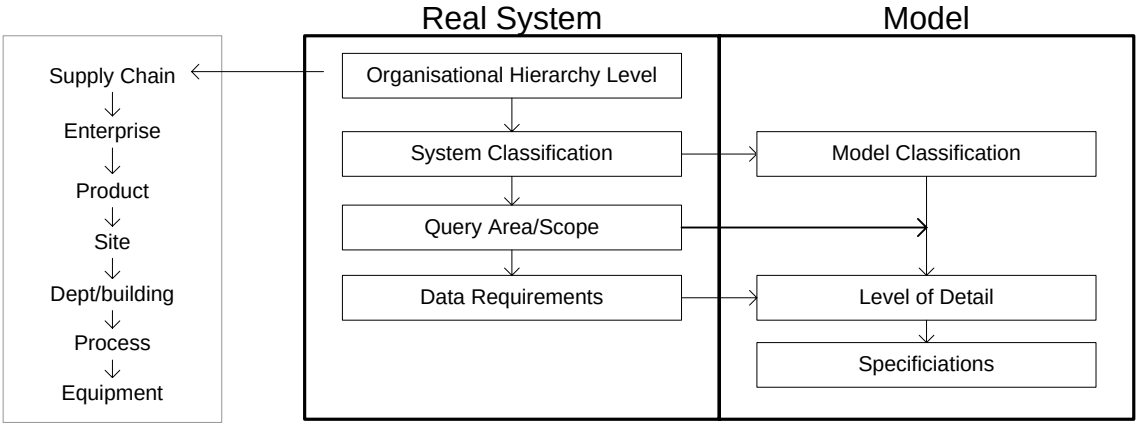


Figure 3.1 Proposed hierarchical framework for domain description and model definition

This work will focus on the 'Process' level of the hierarchy, more specifically on biopharmaceutical batch processes, looking at the manufacture of bulk products. These processes have many features in common which have been listed below:

- RESOURCES (e.g. labour, buffer, equipment)
- ENTITIES (e.g. batches)
- ACTIVITIES

- Product Handling Activities (e.g. fermentation, chromatography)
- Preparation of Intermediates (e.g. buffers)

- Preparation of Equipment (e.g. CIP, SIP)
- Support functions (assays, documentation)
  (Farid et al., 2000)
- Ancillary Activities(e.g. CIP, SIP)

At a higher level of definition, these features are common amongst processes. However if reduced to a lower level, then differences begin to emerge which can impact the way in which these features are modelled. For example a fermentation unit will be similar to a chromatography unit in terms of both having sub-activities and requiring resources such as media or buffer. Although the actual sub-activities or resources will be different, they can be modelled in the same way. However, a chromatography unit will have cycles whereas a fermenter will not, therefore a difference emerges in the way in which the two activities are fundamentally modelled. Differences such as this are the reason for the complexity in trying to standardise the modelling of such systems.

3.3 Scope of Framework

The standard framework described in this work can potentially be applied to any platform and any modelling activity, providing a simple code of practice in approaching model design and construct. The theory behind the framework is that a structured approach to modelling will reduce development time by reducing the likelihood for mistakes in construct and ensure that the client defined scope is fully and relevantly covered, a theory tried and tested in this work. Figure 3.2 shows the route taken.

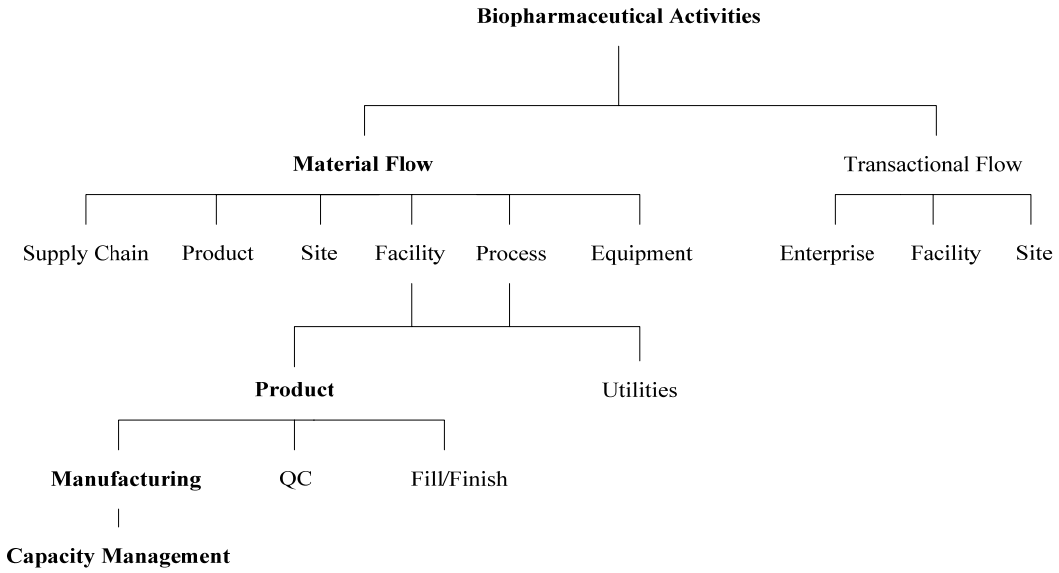


Figure 3.2 Flow chart showing application route taken for the Standard Framework

The reason for taking this route has been two fold. Firstly, with the high cost of manufacture and ever increasing pressures to reduce development time and costs, biopharmaceutical management, in particular capacity management, is where much of the industry's modelling takes place. Secondly and largely due to the former, the case studies carried out during the course of this work were all real client based projects and therefore the nature of each study was inevitably dictated by the client company's modelling needs.

There are a number of capacity management questions which could form the scope of a modelling problem to which the standard framework can be applied. Some examples are given below.

Production Schedule

- When volume exceeds DSP throughput capacity should the manufacturer:
  (1) Scale up DSP to handle full throughput or…
  (2) Increase inventory – store what cannot go through DSP until it's free
  (3) Use more than one DSP. If there are multiple products, do you stagger production to process on a 'first come, first out' basis or pool the products?
- Variations in demand have an impact on the production schedule. If goods are standard should you

(1) Produce to stock in times of low demand to offset capacity requirements in times of high demand?

(2) Produce according to demand and avoid inventory?

Product Mix

- Single product: generate in a single suite according to:

  (1) demand- may mean working below capacity and therefore inefficient utilisation or

  (2) capability - make full use of capacity and store product for times of increased demand

- Multi-product : generate on a

  (1) campaign basis – perhaps according to demand timings. In which case, what should the sequence of campaigns be? What are the campaign durations? What are the campaign changeover costs?

  (2) dedicated production line in parallel. In different suites? Are resources shared between the suites?

Resource Management

- CIP/SIP. There are various questions such as:
  - Take the equipment to the CIP rig or bring the CIP rig to the equipment?
  - Use dedicated CIP rig for each equipment or for each process?
  - Preparation of CIP 'ingredients'?
  - Single-Use or Re-Use (Recycle cleaning solutions)?
- Buffers and other materials
  - Prepare and store for when needed or prepare only when needed? If store, for how long?
- Storage of raw materials

Resource Utilisation

- Changes in utilities utilisation (e.g. WFI for CIP or buffers) after process change/expansion?
- Equipment selection based on

  (1) start from the top and pick the first one that is free

(2) if all busy, which will finish processing first?

(3) based on utilisation – pick the one with the lowest utilisation for balanced approach.

Operational Assumptions

- Pooling
  - Pool products as they arrive then release or operate in a responsive mode i.e. send through on a first come first serve (FIFO) basis?
  - If batching, what is the best batch size?
- Impact of product shelf life/stability on the schedule?
- Shifts, both labour and operational, have an impact on scheduling. For example if equipment must be shut down for the weekend this means that batches must be stored until start up on Monday morning. Therefore operational shifts have an impact on inventory and equipment utilisation. Also, batches must be scheduled such that only those which will be finished before shutdown are allowed to go ahead for processing.

Process Changes / Capacity Constraints

- Fixed capacity scheduling or expansion of facilities / outsource to CMO
- Addition of equipment – what effect will there be on physical space capacity, piping, utilities, CIP access, waste treatment?
- Upstream yield improvements such as increases in titres. What will the DSP effects be? For example on columns?
- Changes in downstream performance i.e. yield/throughput

Disposable versus reusable

- Changes in CIP, waste management, capacity, yield
- How many times can be the disposable be used? i.e. lifetime
- Does the disposable affect the throughput/performance?

Chapters 4-6 describe case studies which look at various capacity management questions. Chapter 4 looks at a case determining how fast a certain number of batches (the demand) can be run through a single product biotechnology facility

taking into consideration resource constraints and process changes. The case study described in Chapter 5 looks at single product manufacture, specifically defining the cycling of batches through downstream process unit in order to maximise facility efficiency (i.e. increase throughput) given constraints such as labour availability and uncertainties such as equipment failure and titre fluctuations. Finally, Chapter 6 considers the introduction of multiple products to the facility described in Chapter 5, looking at the impact of different product changeover procedures and operating shifts on the process throughput.

3.4 Requirement Specifications of a Simulation Model

At the process level, 9 existing DES models were reviewed, all representing various process systems for a large pharmaceutical company including Biotechnology Processes and Logistics, Quality Control Lab Operations, Fill/Finish Operations and Control Rooms for biotech production. Several common elements were found amongst all of these process level models which can be found in Tables 3.1 and 3.2. In this table it can be seen that the features defined under domain description common to biopharmaceutical batch processes have been listed under 'real process features' elements.

In order to truly represent a system it is necessary to model the common elements listed under the Domain Description, that is, all types of activities, resources and entities. Furthermore, in the pharmaceutical industry or indeed any industry with similar activities, there are certain requirements for DES model construction which contribute to the 'Flexible Model Environment'. The features of the flexible model environment form the objectives or requirement specifications of a DES model and there are certain methods available in meeting these objectives using the recognised model elements including the activities, resources and entities.

Following Tables 3.1 and 3.2 are the requirement specifications of a DES model which are platform-independent and state the basic approach to modelling and why that approach is adopted.

Table 3.1 Model elements

Element	Definition
LAYOUT	
Physical Layout	The position of blocks on the workspace and what they represent
Logical Layout	Relates the physical layout of the model to the real world layout of the process, area or network being modelled.
Parallel Activities	Those identical or very similar activities which happen simultaneously.
Main Block Construct	The main functional blocks which form the stream through which the main items flow. Can be activity, equipment or other functional representation depending on the model scope.
Item Transfer	Refers to the way in which blocks are connected and how items move from one block to another.
MODEL INITIATION	
Items Generated / Primary Item	The main items generated and sent through the model. Different to initialisation or trigger items as they are the main items upon which the simulation depends. For example they will hold the important attributes and will in most cases be the model outputs.
MODEL LOGIC	
Metrics	Each model, depending on the nature of the case study, will measure certain parameters. These can be attached to the items, stored in the database or an external file. The type of metrics will in most cases affect how a model is constructed as the measurement of parameters will usually require configuration of blocks and a degree of coding.
Look ahead	The logic used to make decisions based on what is happening in other parts of the model ahead of current time, t.

Table 3.2 Model elements

Element	Definition
REAL 'PROCESS' FEATURES – system constructs which must be mapped to the model construct	
Activities	The components of the system which have time attributes and can be associated with the product and/or resources. These activities can be categorised into Product Handling Activities, Preparation of Buffers/Media, Preparation of Equipment, Support functions and Ancillary Activities
Entities	This is the product units to be modelled. Most likely to be batches.
Resources	Resources can be anything used to perform the model activities for example workers, raw materials, equipment and so on
DATA	
Data transfer	The way in which data required or generated within the model is transferred. Data may be held in the modelling platform or in an external application.
Database	The way in which the database is used and to what extent. Not all models use the database function.
Tracking	The way in which information from one part of the model is used to control another part. It may appear in various forms, for example the tracking of equipment status, the tracking of queue contents and so on.

- *Intuitive to User*

It is important for a model to be clear to the intended user otherwise its usability becomes somewhat diminished. The layout of the model along with the hierarchical structure, the interactions between blocksand nomenclature should make sense to the intended user. The model should also be representative of the physical or logical nature of the system being modelled.

In their paper Valentin and Verbraceck (2002) propose several guidelines for the design of a model to overcome the problems in complex simulation studies by introducing reusability and maintainability. They illustrate this using a case for passenger modelling at airports however the main principles can be applied to modelling of any system. Three of the guidelines are of particular interest here:

'Interactions between model parts…should represent interactions in the real system'

Comprehensibility for the user of a simulation model depends greatly on the user identifying with the model components. In order for this to happen the model must have interactions between components which clearly represent interactions between objects in the real system. Interactions between model components are related to item transfer which is the way in which blocks are connected and how items move from one block to another. Any material/item flow should be clearly defined and consistent. The item itself should be one that the user is familiar with e.g. batch of product, pallet of material, document, worker and so on. Resources should also be used and tracked in a manner that makes sense to the user.

'Use concepts that represent functionalities as found in reality and that can be used for visualisation purposes'

The functionalities in reality can be defined as the activities which take place in the real system and whose representation is required to meet the first and third guidelines. Linking directly to the first guideline, this provides an intuitive model by creating a true representation of the real system. In other words the functions in the real system within the scope of the case study should be defined in the model. Existing models reviewed thus far have illustrated this characteristic by modelling activities as the main blocks construct, defined as the main functional blocks which

form the stream through which the main items flow. That is, the blocks on the top most level of the model which are the main hierarchical blocks. Block constructs can be activities, equipment or other functional representation depending on the model context.

'Visualise a system in such a way that complexity is reduced but the essential processes are still shown'

In order to create an intuitive and useful model it is important to only model the most essential components of the real system. Those activities which have no bearing on the model input and outputs will not add value but only complexity, also adding to model run time.

- *Relevance and Ease of Data Input/Output*

Linking directly to the last guideline above the model should have the sufficient complexity and data input to be relevant and useful to the user. It is also important that the user be able to run scenarios without modifying the model structure and therefore all user input parameters should be accessible without going deep into the model. This links also to the importance of model output; each model, depending on the nature of the case study, will measure certain parameters. These can be attached to the items, stored in the database or an external file and may be used to plot graphical representation within package (if that functionality exists) or exported to a package such as Excel for manipulation. Whichever method used, retrieval and storage with scenario description should be straightforward and without the prerequisite for detailed knowledge of the model.

- *Short Run Time*

Firstly let us imagine two types of model complexity. The first, visible complexity, comes about from the use of too many blocks within one model. The second, hidden complexity, is the use of underlying discrete event coding as part of the model logic. In a perfect world, a model should take a matter of seconds to run. In order to reduce the run time the visible complexity must be reduced i.e. the number of blocks in the model which must be executed during a run. This of course means that in some cases, longer and more complex coding is required to maintain the functionality of

the blocks thus increasing the hidden complexity. As a result this in most cases will somewhat diminish the intuitiveness of the model along with reusability and sustainability by anyone other than the creator of the model. Therefore a balance must be created between the need for fewer blocks and feasible coding.

- *Maximised Reusability and Sustainability*

Although most models are created to address specific issues reusability of a model can be important especially with rapidly evolving systems. If a model can be reused for a system which has gone through various degrees of change then it will negate the need for model rebuild such as through the use of standard building blocks and nomenclature. Furthermore it is anticipated that similar issues could be addressed in related business units, and so models should be constructed and developed in a way that would make model transfer to a related area efficient.

- *Minimised DevelopmentTime*

Starting new models from scratch is very time consuming and typically the model developer is under a time constraint to meet the business needs. Therefore there should be a structured approach to the model development that minimises development time and the need for user input (by focussed approach). Once the effort has been put into modelling a process, to maximise the return on that investment the model should be capable of further use, and ideally incorporated into the business of running the process (e.g. business planning, change control, improvement projects including 6σ).

This supports some of the previous requirements in that the model must be usable by non modelling users.

To help ensure models are reusable, maintainable, and suitable for future development the use of "custom code" should be minimised. Though the use of discrete event code can significantly help streamline models both in size and run time, it adds considerably to the difficulty in maintaining models, and trying to understand how they work thus a need for a trade off between use of code and model size.

3.4.1 Meeting the Requirement Specifications in Extend

Tables 3.1 and 3.2 summarised the elements of a DES model along with a brief definition for each. This list of model elements was compiled following a review of eleven existing models, constructed by modellers at an academic institute (UCL) and at a multinational pharmaceutical company. There were five categories of elements given: Layout, Model Initiation, Model Logic, 'Real Process Features' and Data. Table 3.3 shows the same list of model elements and the correlation of each to the requirement specifications.

It shows that if the structured framework is applied to the modelling process then each element will contribute to meeting the requirements in one or more ways i.e. it maps the relevance of each element to the requirement specifications. For example, the standard framework will give guidance on the modelling techniques to be used in resource utilisation which will lead to a lower number of blocks being used. This will contribute to the intuitiveness of the model.

The following describes how each of the elements can contribute to the meeting of one or more of the six requirement specifications.

Layout

Physical Layout / Main Block Construct

The physical layout of a model includes the arrangement of blocks on the workspace (the white area on which the model is built), the hierarchical structure and what the blocks represent.

DES modelling in Extend involves hierarchal levels containing blocks. For an intuitive model it is necessary to reduce complexity at each level. IDEF0 describes the Top-Level Context Diagram where the top level of a model contains a single box to represent the subject of the model. This is useful in giving the model context, especially at the lower levels of the organization hierarchy.

The standard also refers to the sub levels of a model as Child Diagrams, whereby functions are decomposed into their sub-functions. According to the standard, the number of child boxes on each child diagram should be limited to 3-6.

Table 3.3 Summary of Model Elements and Contribution to Requirement Specifications

Element / **Requirement Specification**	Intuitive to user	Relevance	Ease of data input/output	Short run time	Maximised reusability and sustainability	Minimised development time
Physical Layout	●					
Logical Layout	●					
Parallel Activities	●			●		●
Main Block Construct	●					
Item Transfer	●					
Items Generated / Primary Item	●	●				
Metrics		●	●			
Look ahead		●			●	
Entities	●					
CIP/SIP	●			●	●	●
Resources	●	●				
Data transfer		●	●			
Database			●			
Tracking	●			●		

Finally IDEF0 describes three types of decomposition: Functional Decomposition (breaks down activities according to *what* is done), Role Decomposition (breaks down things according to *who* does what) and Lifecycle Decomposition (breaks down a system first by the *phases of activity*). The former two are the most common methods in bioprocess modelling however the model construct elements such as resources, item tracking and so on differ quite significantly between the methods of decomposition. Imagine a granulation system whereby batches are dispensed into bins, granulated, and then sent through to compression. According to IDEF0 there are two ways in which the system would be modelled:

Method 1: Functional decomposition whereby the main block constructs would be the activities: 'granulation', 'dispensing' and 'compressing' with the actual equipment held in resource pools.

Method 2: Role decomposition whereby the main block constructs would be the equipment: 'granulator', 'dispenser' and 'compressor'.

At the process level functionalities found in reality should be represented in the model as the main block constructs i.e. Functional Decomposition, for both visualization purposes and for resource management. Using this method the availability of equipment can be easily tracked without the use of global arrays (these are essentially matrices which hold information within the model), thus reducing the number of blocks necessary for the running of the model. Furthermore modelling main objects like equipment as resources will allow for the externalisation of ancillary activities such as CIP and SIP, a beneficial modelling method as described in a later section.

Logical Layout

The logical layout links the physical layout of a model to the real system. In order for a model to be intuitive and for a user to relate to what he/she is seeing on the screen, it is best to logically place system components as they would be relative to each other in the real system. When the main process chain is modelled this logical layout should be naturally built-in, with a left to right convention used as the flow direction (unless backward flow is an integral part of the system/model).

For an intuitive model functions should also ideally be hierarchically modelled together. E.g. if two pieces of equipment / activities perform the same function, there should only be one hierarchical block representing that function to contain them; a hierarchical block is a block which contains sub levels. Those on the top parent diagram are the functional hierarchical blocks as they represent functions.

Parallel Activities

When identical activities occur in parallel, in order to reduce model visible complexity and size it is usually desirable to compress them much in the same way as modelling the same functions under one hierarchical block. Furthermore if less cutting and pasting is required to duplicate the activity blocks there will be a reduction in the time taken for constructing the model, albeit a small reduction. Importantly, however it is not possible to compress parallel activities when lookahead logic requires separate activity blocks (for example, unique and distinguishable queues required to sit within the blocks).

Item Transfer

The IDEF0 standard states that boxes must be connected by conventional solid arrows. This representation of components and their links results in an intuitive model as flow is clearly represented. Within the Extend environment it is possible to use arrows or simply straight lines to link blocks. There is very little difference between the two methods and if the left to right convention has been adhered to, the direction of flow will be apparent without the use of arrow heads.

Model Initiation

Items Generated / Primary Item

The primary item should represent the key physical entity being modelled (and thus be relevant), often in manufacturing this will be a batch or part of a batch (section). Tracking items should be relatively easy i.e. none or very limited use of catch/throw blocks. This is due to the fact that these blocks have no visible connections between them and can be placed in any window anywhere in the model. If an item is thrown across windows, it is near impossible to track its movement thus making debugging extremely difficult. The ability to visually track the movement of items also creates a more intuitive model.

Model Logic

Metrics/Outputs

The metrics will depend on the case study or scope of the model and will therefore vary. In those models using equipment, utilization seems to be a common metric and therefore may be a constant associated with all equipment based models. In terms of viewing the data collected there are two methods:

Method 1: within the Extend model during or after a run

Method 2: within an external application such as Excel

Many of the models analysed used graphs at various stages within the model to represent metrics such as resource utilization i.e. method 1. Placing the plots next to the corresponding pools, equipment and so on seems to be convenient for simple and immediate analysis. However it is ultimately best to output all data into a separate application such as Excel for more complex data manipulation. As the requirements of the metrics is variable according to the model and case study, sending all data to an external file allows for categorisation of outputs according to scope, thus allowing a single model to be used for various capacity management questions. For example, if a data sink file is created in Excel, all attributes and data collected during the run can be sent to that file. Ultimately, using method 2, different combinations of data can be copied to pivot tables, each corresponding to a different scope, extending the reusability of the model.

Look-ahead

This is a modelling concept which is actually quite complex. Imagine a Fill/Finish model for a freeze drying process, whereby the dryer needs to be prepared for unloading its contents 24 hours before it has finished processing them. The simplest method of modelling this would be to subtract 24 hours from the processing time and to then create and send a trigger item to enter the unload preparation activity blocks. However a far more sophisticated method of lookahead logic is required whereby it may be used in circumstances where:

- processing time is 0 or less than the ‘prepare ahead’ time
- the processing time is unknown
- the look ahead is not dependent on processing time

Two possible methods of a creating a more flexible and generic look-ahead would be:

Method 1: Run a parallel logic which looks up all the future processing times (if available), and minuses the 'prepare ahead' time from that to give the start time of the trigger activity. This information is then conveyed to the main process chain.

Method 2: At some point early on in the model a discrete event equation block is used to calculate the time at which the preparation of the dryer should begin by calculating the current Extend system time, adding the dryer processing time to that and subtracting the preparation time from the result. This gives a system time for when the dryer should be prepared for that item. This information is sent to a created database schedule table which is used for the generation of the trigger item.

It is believed that method 2 or a similar variation should be used. Firstly the main problem with method 1 is that the additional logic running in parallel increases the model run time. Secondly method 2 is a highly generic method which can be used for any model (the input times can easily be changed), when the processing is less than 0 or the 'prepare ahead' time and when the logic is not based on the processing time. Both methods fall where the processing time is not known. Method 2 however can be adapted to include logic which calculates the processing time instead of just referencing the data source.

Real Process Characteristics

Entities

Entities are those objects in the modelling environment which represent people or items in the real system. The most common entities, other than the primary items, are labour and there are various ways of modelling labour in Extend and the choice should depend on the nature of model metrics and the scope of the case study. For example

Method 1: If labour is a constraint or utilisation is a metric then a labour block can be used. This is an inbuilt Extend function which allows for the allocation of attributes and costs based on labour usage. Furthermore, as in the case of a resource pallet, this block is an 'item through' block which means that labour is modelled as actual items. Any constraints can be represented physically by stoppage in the model due to unavailable labour.

Method 2: Labour can also be modelled in resource pools whereby they are not items, simply units of resources with attributes and utilisation figures attached, the main difference being that allocation can be based on difference conditions set within the resource pool block.

Due to the fact that labour does not usually have associated with it activities independent of the primary items there would be little added benefits to modelling them as physical items within the model. Therefore it is best to use method 2. The constraints due to labour shortage are modelled much in the same way as method 1 i.e. if the labour is not available the item requesting it will simply sit in a queue. However in addition the resource pool gives the extra option of labour allocation based on first come first served or given priority.

Ancillary Activities (CIP/SIP)

Cleaning procedures only correspond to equipment used either within the main process stream or for ancillary activities such as buffer preparation and waste disposal. There are two ways in which CIP/SIP can be modelled and a number of reasons for choosing one method over the other.

Method 1: As part of the main process chain logic. This means that a block(s) representing CIP/SIP is placed wherever the activity is required. For example in hierarchical block A representing fermentation there may be 3 further sub-functional blocks in the child diagram. These may represent filling, fermentation and harvesting. The latter will be directly followed by a clean and a CIP/SIP block will be placed there accordingly. This method means that there may be multiple (depending on the number of clean occurrences) CIP/SIP blocks and is used when the equipment are modelled as the main block constructs (role decomposition).

Method 2: The CIP/SIP will be represented by one single block which will be externalised from the main process chain. This method is used when there is functional decomposition and the equipment are modelled as item resources. When cleaning is required the equipment is sent to the clean block and returned to its pool/pallet or back to the process stream. In order to model multiple equipment being cleaned at the same time there are two scenarios to be considered. The first is if all equipment share the same CIP/SIP rig. Here a queue should be used to represent the wait time. In the second scenario, more than one rig is available, in which case the

same CIP/SIP process stream can be used by the different rig/equipment items, representing the overlap of the cleaning procedures.

This method means that the requirement specifications are more fully met for various reasons. Firstly it reduces the number of building blocks (visible complexity) and thus allows for a degree of reduction in the model run time. Secondly it creates a more intuitive model in the case where there is a CIP/SIP equipment or material constraint as it allows for visual tracking of item movement through clean. Finally if a generic CIP/SIP is built, it will contribute greatly to the maximisation of model reusability and sustainability.

Resources

Resources such as materials, equipment, utilities and labour can be modelled in a number of ways.

Method 1: modelled as an item. This is beneficial when it is necessary to visually track the resource's movements, when the resource and primary item need to be paired for a section of the model, and/or the resource carries its own attributes. The resource item can either be generated then discarded or be managed via a "pallet block".

Method 2: modelled in a resource pool where the aspect of the resource is the number available at any specific time. One pool may be used to hold all similar resources if individual utilisation data is not required or it is calculated elsewhere and/or usage is based on the same shift/rule or this is determined elsewhere. This method is to be used where the conditions described under method 1 do not exist such as in the case of labour. Furthermore, consumable resources such as water or buffer materials should be held in resource pools as they do not loop back to be used again (unless recycling of such materials is part of the system being modelled).

ISA-88 describes the way in which resources should be allocated and arbitrated. For the purpose of model construction resource allocation and arbitration is an important factor. Extend allows the use of blocks such as Select Output which makes path decisions (and therefore resource decisions depending on model setup) based on toggle or input values. A selection criteria such as 'select equipment based on input

volume' can also be achieved however this requires some slightly more complex configuring of the inbuilt blocks.

In terms of arbitration there are ways of dealing with resource demand in Extend, for example, prioritising resource demand. Also the left to right logic of the program means that in the process path, the farthest left will use that resource first.

An algorithm such as "first come/first served" might be used as a basis for arbitration' and can be seen in Extend in the form of FIFO (first in, first out) queues.

Data

Data Transfer

As stated under the overall requirement specification there must be sufficient complexity and data input to be relevant and useful to the user. It is only necessary to have user defined inputs where they will fall within the scope of the model. In other words the inputs required are those which allow the model to calculate/output parameters such as cost, time, yield/throughput and resource utilisation. Input parameters should also be accessible without going deep into the model. There are two ways of achieving this

Method 1: input parameters can be entered into the database, either in Excel or Extend. The problem with the latter is that the user will be introduced to the underlying data source of the model.

Method 2: by creating a notebook level where all input locations are cloned from block dialog boxes. The user will only see the notebook containing a list of inputs and their meanings. Outputs can be cloned onto the notebook in the same way. This method is simple and user friendly, and does not require the opening of additional files, thus making it the better option.

Database

The Extend platform uses an interface between Excel to transfer data to and from its inbuilt database. The best way to enter and manipulate the data is to first work in Excel. As a data handler the Excel software is much more equipped and user friendly. Furthermore it is more or less a universal software and therefore almost all users will have access to and knowledge of it (for purposes of future model changes). The data can then be imported into Extend and used for the running of the model.

However difficulties arise when changes are made to the database in both Excel and Extend. It is important that the two be synchronised, or better still, for all changes to be made in Excel.

Tracking

There are two ways to track items or values as they change within the dynamic system (model).

Method 1: Global Arrays. These are basically matrices which hold real or integer values and can be built to specifications in terms of columns and rows. Values corresponding to item, resources etc can be held in global arrays, updated periodically and accessed either by replicating the array block itself and connecting it to the block reading the value or by using code to look up the matrix address (array index, row index and column index). The latter helps in reducing the number of blocks needed if more than one address is required for decision logic.

Method 2: by using physical items much in the same way as using item resources. Visual tracking of items is often useful in cases of debugging and can contribute to the intuitiveness of the model by allowing the user to see exactly where items are at any one time. The two methods of tracking are not mutually exclusive and a combination of both is necessary to build models representing complex systems with many interactions.

3.5 Evolution of Standard Framework

This section will discuss the standard framework in its final version and how it evolved over time, looking at the shortcomings of each of the earlier versions and how they were highlighted through application to biopharmaceutical manufacturing case studies.

3.5.1 Overall Structure

3.5.1.1 The Evolutionary Process

Three versions of the framework were developed. Figures 3.3, 3.5 and 3.8 show the overall structures of the different versions while figures 3.4, 3.6 and 3.7 show the timelines for the corresponding modelling case studies to which they were applied.

As figure 3.3 shows, Version 1 of the framework consisted of scope definition, model characterisation and model construction. As the timeline in Figure 3.4 shows, the model took 7 months to build at the end of which a rebuild was necessary, thus extending the timeline even further. In fact the overall duration of the BioSynT case study was approximately 9 months. Section 3.5.1.3discusses the reasons for this.

Figure 3.5 shows Version 2 of the framework which expanded on the Problem Structuring phase of Version 1 by adding a non-coded description of the system. It also added a Design Phase, a model specific description of the system, mapping the system or process elements to the model elements. As the timeline in Figure 3.6 shows, the overall mAb case study took just under 5 months, with the actual model construction phase significantly reducing from 5.5 months to only 1 month. However, the debugging stage did take longer than expected with a duration of 3 months. Section 3.5.1.3discusses the reasons for this.

Figure 3.8 shows Version 3 (final version) of the framework, which is very similar to Version 2 with the only difference being the addition of the templates. The library of templates were created to reduce not only the model build time but also the debugging time, by providing building blocks which could easily be used and debugged due to their standard nature. Figure 3.7 shows the projected timeline if the mAb case study were to be carried out again using the framework Version 3, showing that the overall duration would significantly reduce by 65%.

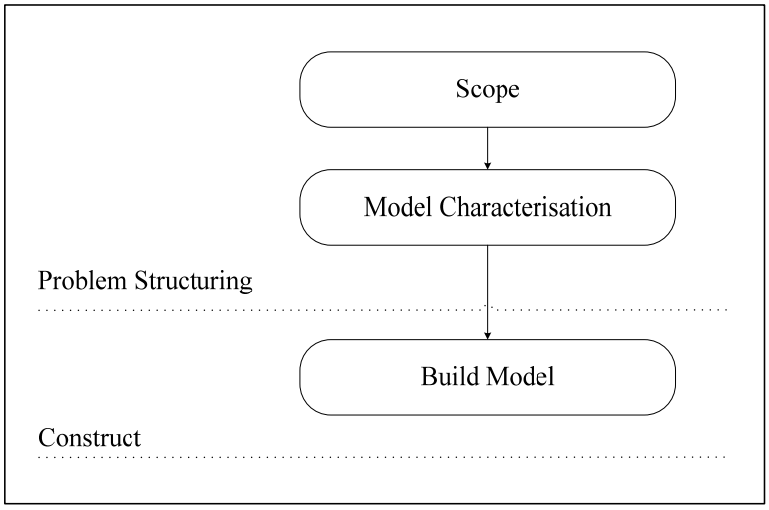


Figure 3.3 Proposed methodology for the different stages of model design and construct as part of the Standard Framework Version 1

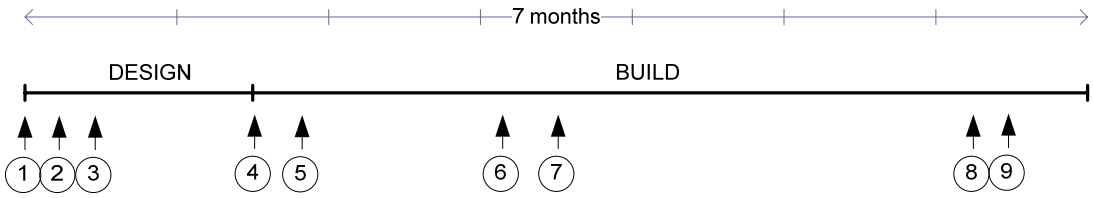


Figure 3.4 Stages of BioSynT model build (Case study for Standard Framework Version 1).

Where

1. Given remit with a series of Gantt charts. No direct contact with client

2. Independent review of possible capacity management questions that could be asked

3. Began design of model based on predetermined standards and guidelines such as decomposition, resource modelling and data management

4. Began model construct based on standards and limited information given with remit.

5. Structure of model put into place. Initial elements added such as activities, database for data management, resources and cycle times.

6. First contact with client. Scope increased but no further information given.

7. Increased levels of complexity added to account for increased scope, largely based on assumptions. A few rules on resource usage added based on discussions

8. Second meeting with client. Some data provided - new information suggests that model contains too much unnecessary/redundant complexity based on previous assumptions. Decision point on how to proceed: revisions will mean more code, more rules and more data to manage, resulting in high levels of hidden and visual complexity. Model rebuild will mean more time dedicated to model construct however expected increase in model efficiency with new knowledge.

9. Model REBUILD

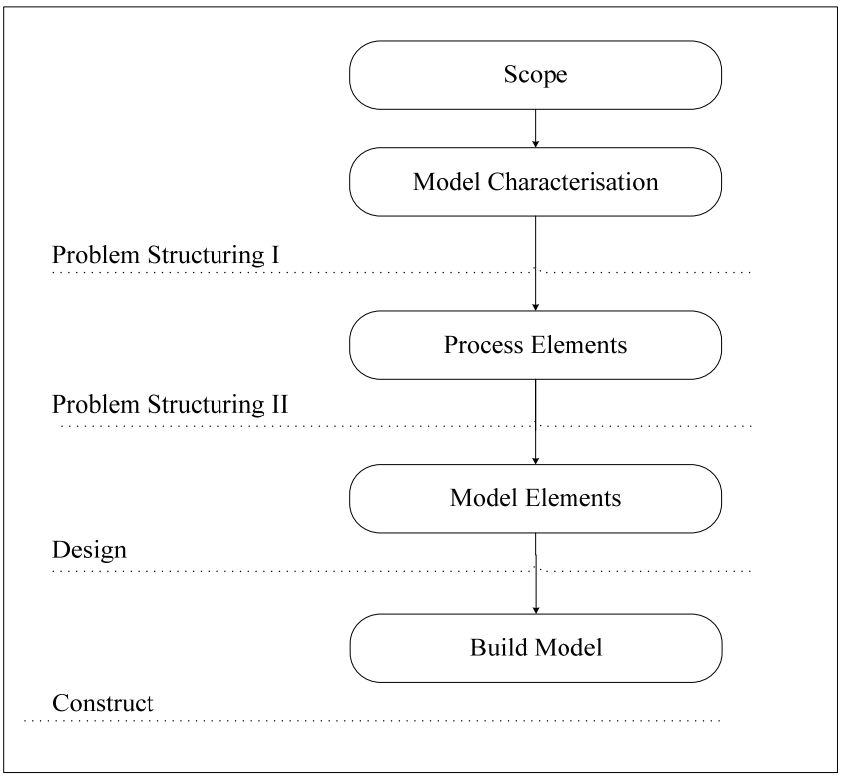


Figure 3.5 Proposed methodology for the different stages of model design and construct as part of the Standard Framework Version 2

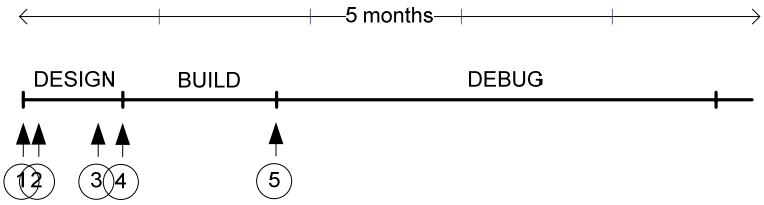


Figure 3.6 Stages of mAb model build (Case study for Standard Framework Version 2)

Where

1. Problem Structuring Phase I (2 days)
2. Problem Structuring Phase II (14 days)
3. Design (4 days)
4. Build (30 days) – including early version templates construct
5. Debugging (90 days)

(Validation occurred throughput the project with the help of system experts)

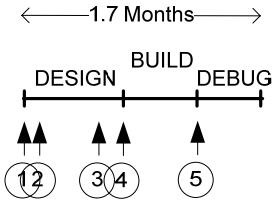


Figure 3.7 Projected Stages of mAb model build (Standard Framework Version 3)

Where

1. Problem Structuring Phase I (2 days)
2. Problem Structuring Phase II (14 days)
3. Design (4 days)
4. Build (15 days)
5. Debugging (15 days)

(Validation occurred throughput the project with the help of system experts)

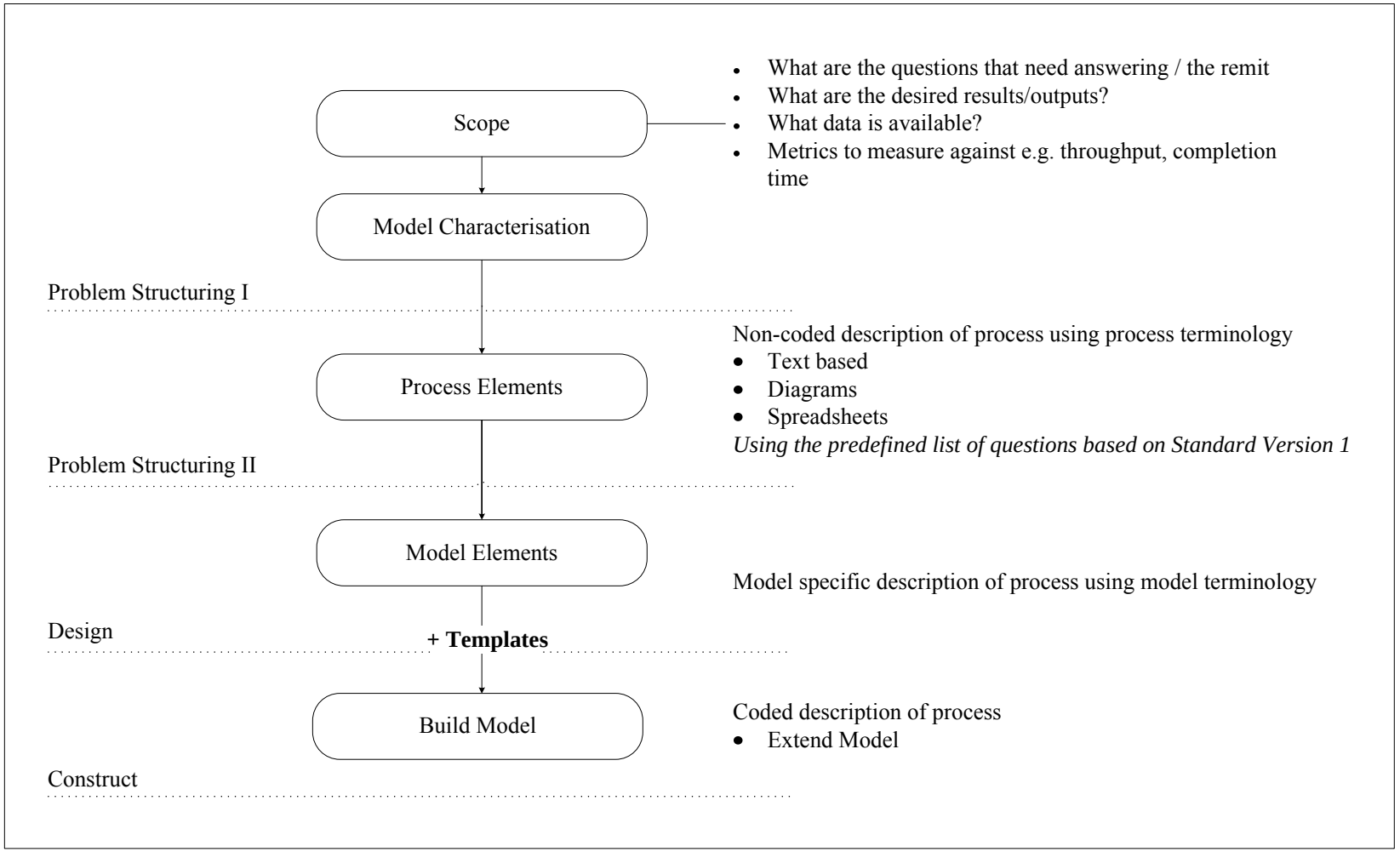


Figure 3.8 Proposed methodology for the different stages of model design and construct as part of the Standard Framework Version 3

3.5.1.2 The Standard Framework Final Version

Since developed as part of an evolutionary process, Version 3 bares a similar resemblance to the first two versions, but with additional features such a more comprehensive methodology and a library of template blocks. Furthermore, a significant finding of this work is that in order to minimise the degree of customisation required the standard, and in particular the templates, should be created with a certain degree of inbuilt customisation. This move away from a completely generic framework inherently limits the number of systems for which the templates can be applied. As such, although the modelling methodology is a universal one and can be applied to any modelling activity of any system, the majority of the templates will be limited to the specific domain of biopharmaceutical manufacture. The following describes the key features of the standard framework Version 3.

The final version of the standard framework consists of three major parts:

(1) Methodology for model development

(2) Standard templates for model construct

(3) Standard set of questions to ask the client

3.5.2 The Methodology

There are two possible approaches to the development of a model 1) begin construct immediately, or 2) design on paper before even approaching the modelling platform. It is believed that the latter method of approach – design before build – will increase the efficiency in terms of block usage and will importantly reduce the amount of model construction time (Robinson 2008).

The methodology proposed here is a series of steps defined to provide a more structured and disciplined approach to model development by clearly defining the problem formulation and model design stages. Its aim is to reduce the time taken for model development by

- speeding up the data gathering process

- ensuring a clear and well defined scope which will lead to a more relevant model
- providing a way for the developer to more easily and quickly verify their understanding of the process before construction

Brooks and Robinson (2001) define two stages prior to the construction of a simulation model, the 'Problem Structuring' stage where the problem and system are clearly defined, and the 'Conceptual Modelling' stage, defined as a software independent description of the model that is to be constructed. Here these two stages have been combined under Problem Structuring Phases I and II.

Problem Structuring I

Scope

This is given by the client or the user and describes the questions that they would like answered in the model. For example, in the BioSynT model, the scope was to determine how fast the process could be run in series mode given a series of constraints.

The scope will also cover the inputs or the data available and the desired results or outputs that the client would like to see. For example, in BioSynT, they were interested in seeing the number of tickets generated during the process and were able to provide a spreadsheet of the ticket generating tasks.

Finally, the scope will cover the metrics that the client would like to measure against, for example, production costs, throughput or completion time.

Model Characterisation

The scope will lead to the definition of the type of model required to answer the questions. This definition or characterisation will describe the platform and the system. For example the BioSynT case study considered a scheduling problem, looking at the dynamic utilisation of resources for the production of a product with no consideration of any mass balancing. Therefore the model required was characterised as an 'Extend Manufacturing' model, based on the selection criteria described in Chapter 2.

Problem Structuring II

Process Elements

These are used to create a text-based description of the process and are the elements which were defined under Section 3.2. Used as a form of questionnaire, giving a more structured approach to the developer/client interaction and providing a means of obtaining the relevant information needed to create an accurate picture of the system. The terminology used here is process terminology so that both developer and client have a clear understanding of the outcome.

Furthermore it is also useful at this stage to identify those parameters which are variable so that this variability can be added to the model and controlled by the user.

Design

Model Elements

Here the process elements are used to design the model, considering the various modelling options to map them. The terminology used here is model terminology, translating those used under process elements. For example, tasks become activities.

Construct

Build Model

This is where the process elements are actually mapped onto the modelling platform using templates.

There are four types of template as shown in Figure 3.9.

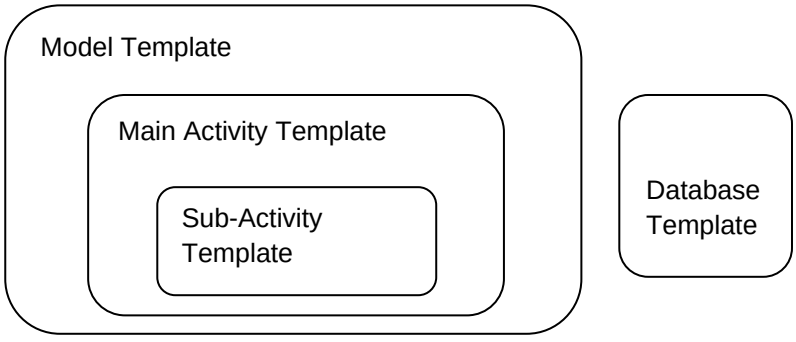


Figure 3.9 The different types of model template as part of the Standard Framework Version 3

The database template provides a predefined set of tables which the developer can fill in. Its structure follows closely that of the questionnaire given in Table 3.4 with all the common system elements incorporated.

The model template is a generic structure template which acts as a guide, telling the developer where elements should be placed by containing hierarchical top level blocks.

A combination of the Main Activity Template and the Sub-Activity Template has been used to create a standard template with a certain degree of customisation, with those elements specific to the system later added by the model developer. So this means a main activity template contains various built in elements as found to be common amongst the different systems such as the gates, equipment and also the sub-activities.
The sub-activity blocks themselves have been given a standard structure; all sub-activity blocks are identical and are given identity by entering a single number in a clearly defined box. The first sub-activity is number 1, the second number 2 and so on. Therefore the nature of the actual sub-activity e.g. wash, need not be known. The identifying number is used by the block to reference the correct tables in the database for all required parameters.

Figure 3.10 shows the different types of template which can be used at different hierarchical levels to build a model.

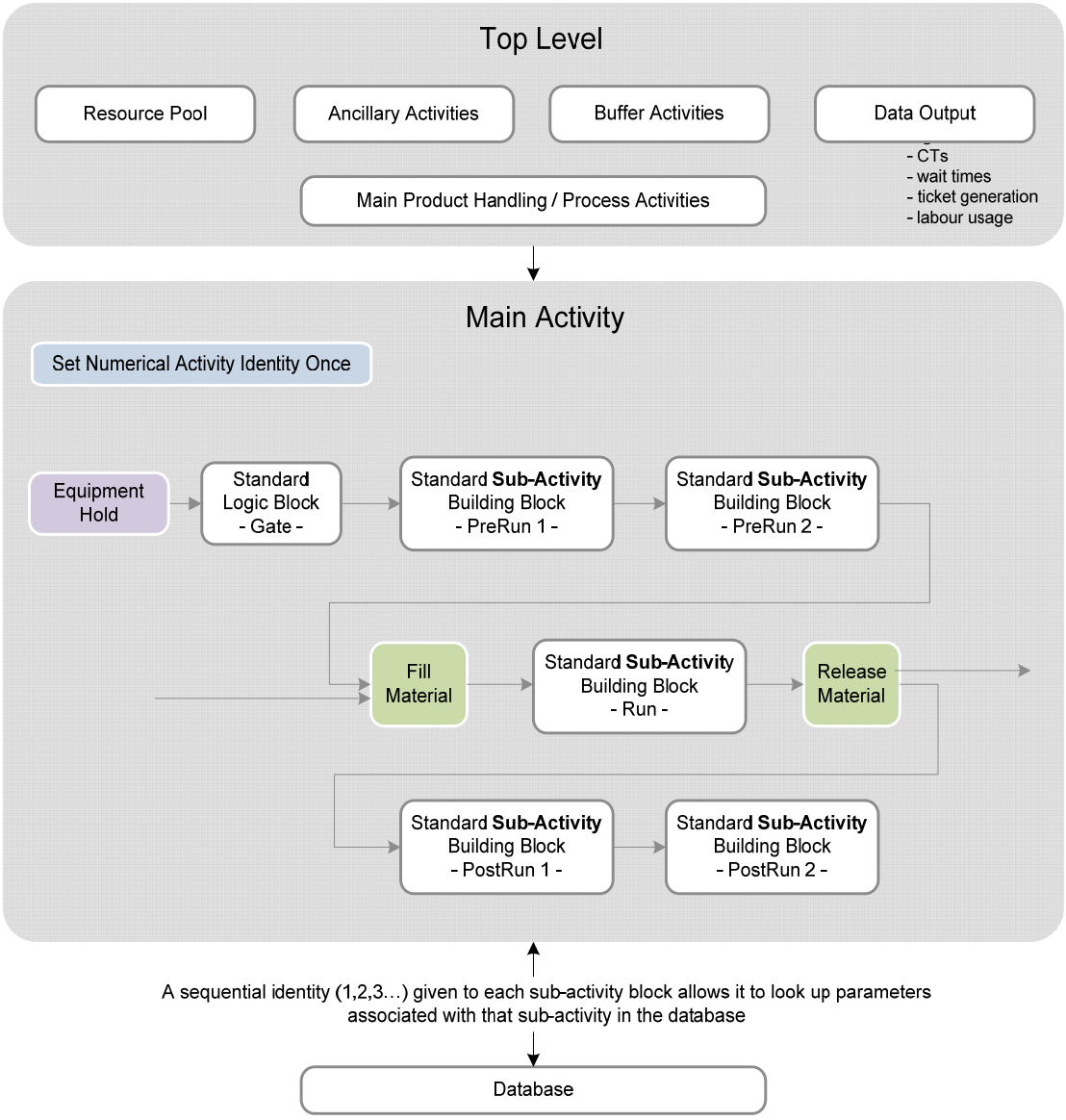


Figure 3.10 Different templates created under Standard Framework Version 3

3.5.2.1 Support for the Standard Framework Version 3

The following sections describe in greater detail why the framework evolved as it did, giving the reasons behind the additions made.

Problem Structuring and Design

The two phases, Problem Structuring II and Design, were omitted from the first version of the standard framework. As a result, when the standard was applied to the BioSynT case study, the project took far longer than expected and required a complete rebuild because it simply did not meet the requirement specifications and

failed to fully cover the given scope (albeit a rather unclear scope). This is shown by the timeline in Figure 3.4.

The first problem was that the model was built too quickly based on too many assumptions where the information required was not available. In fact an initial meeting with the client did not take place until the majority of the model was already in place, having been built based entirely on a process Gantt Chart. The problem faced here regarding the data for modelling is not a new one. Sadowski and Grabau (2000) suggest that the problems regarding data are that it can be insufficient or at times excessive such that the modeller has difficulty in identifying the relevant parts and furthermore there can be the danger of misinterpretation of the data especially where the modeller is unfamiliar with the system.

Secondly, many important system elements which greatly affect the scheduling and resource management were omitted such as probe failures, additional rules like storage allowances and wait thresholds on columns. One could argue that these are not added upon initial model construct anyway and only come about after discussions with the client. However this did not happen. Such elements may not be discussed on the first or even second meeting with the client for two reasons, (1) the client is usually unaware of the capabilities of the model and the significance of these elements to its running (2) the modeller is more often than not unfamiliar with the system and therefore does not know the right questions to ask or data to seek.

What these reflections on the initial model construct suggest is that the approach to the process may have been wrong and that greater or, more importantly, 'better' communication with the client would have shortened the construct phase and would most likely have negated the need for an entire rebuild. Here is posed a dilemma: a hasty model construct can result in a model largely based on assumptions and not truly representative of the system. But too much time spent on the design and waiting for data can delay construct indefinitely. The solution here is to have a set of standard questions which are common to similar systems. For example perhaps with a biotechnology manufacturing system the questions must always be regarding Cycle Times, Activities, Resources, Rules for resource usage and Labour. This would help in the case of a model where the remit is to look at scheduling. But what about in the case where the model needs to answer the question of flow, taking into consideration utilities availability and flow constraints in the same biotech facility? In which case

information such as Vessel Volume, Flowrates and Utility requirements would be more useful than labour and perhaps resources such as buffers (unless these too share the same utilities). Therefore it is not only the system being modelled which must be considered when asking for the relevant information but also the question being asked or the remit.

Templates

These observations made above were tested during the mAb case study, which as Figure 3.5 shows, took significantly less time than the BioSynT study. This can not only be attributed to the existence of a far more comprehensive problem structuring and design stage but also the existence of the early version templates which aided in the construction process by acting as basic building blocks, reducing the variability in structure and narrowing down the possibilities amongst the myriad of ways to model a particular system element or feature. However, the mAb case also showed that there are shortfalls to the framework. Firstly, although the construction phase was accelerated, the debugging stage took far too long. It can be argued that better model outputs or built in indicators could have helped. Secondly, the degree of customisation later made to each of the template blocks was quite significant, with many features such as lot cycles added to the chromatography columns. Perhaps the templates could be taken a step further by creating different blocks for the different functions such as fermenters and columns. This would subsequently reduce the degree of customisation and thus reduce the construction time, possibly impacting the debugging phase also.

The database template was initially designed and built within Extend, using the SDI link to externalise it to Excel for user input. However a major flaw with the SDI tool is that in order for Extend to be able to read the external database, its structure must follow certain rules, which do not lend themselves to a necessarily user friendly input. In this case, upon review with the end user of the model, it was decided that bypassing the SDI tool would create a far more intuitive database, with structures familiar to the system users. Subsequently the database was externalised, making the Excel end of the data link the front end of the database construct, with all table structures designed to maximise user intuitiveness. The data from this database would then feed into the internal database automatically upon model initialisation,

taking the form required for global reference within the model. Section 3.5.3 will describe this template in far greater detail.

Due to the fact that the templates library very much evolved during the construction of the mAb model, additional template blocks were created to meet requirements of system features as they arose during the project validation meetings. Upon review, it wasseen that many of the blocks could be combined. For example, there were three different blocks, the Makeup, Primary and Secondary activity blocks, which were practically identical in structure but which looked different and had different names. Combining these would simplify the library significantly.

Conversely, the level of standardisation created by building a single activity block for all system functions is perhaps too great. For example, the single main product handling activity template block was used to build all unit operations from fermentation to chromatography. The difference between the operations was then built in by tailoring the blocks to represent the system elements. It can be rather difficult to strike the correct balance between standardisation and customisation, with general consensus stating a 80/20 rule appropriate. That is, 80% of the functionality is provided by the template and 20% is customised by the modeller to achieve 100% system representation. However the degree to which these blocks were changed or added to far exceeded 20% in this case and therefore it can be argued that the level of standardisation was too great. For example, perhaps it would be more efficient to create separate blocks for the different unit operations allowing for their different system features to be incorporated into the template rather than later added in.

This increased specificity of each functional block poses a potential problem as it could limit the use of these blocks across different types of biopharmaceutical system. For example, could the library blocks be used to rebuild the fill/finish model? The answer is most likely no, as the level of customised functionality added means that the relevance of each block to a different system will be diminished. In order to be representative, each block would have to be further customised, exceeding the 20% desired customisation threshold, thus making model construct inefficient. Therefore, it is proposed that there be a different library of blocks for the different systems i.e. one for biopharmaceutical manufacturing and one for fill/finish. This grouping of manufacturing systems is made possible by the commonalities

between the systems, for example the existence of chromatography columns in both the BioSynT case and mAb. Furthermore, the different types of activities found are also common, for example, all biopharmaceutical manufacturing processes will have ancillary activities and buffer/media makeup of some sort. Whether these elements fall within the scope of the model is at the discretion of the modeller however they will be present in the library if needed.

3.5.3 Detailed Structure

This section describes, in detail, the features of the proposed framework. The methodology which forms the premise of the framework consists of problem structuring, design and construction phases along with the use of generic building blocks. These are all based on certain inputs and outputs found to be common among biopharmaceutical manufacturing systems, in particular, within the scope of capacity management. Figure 3.11 shows these input and outputs.

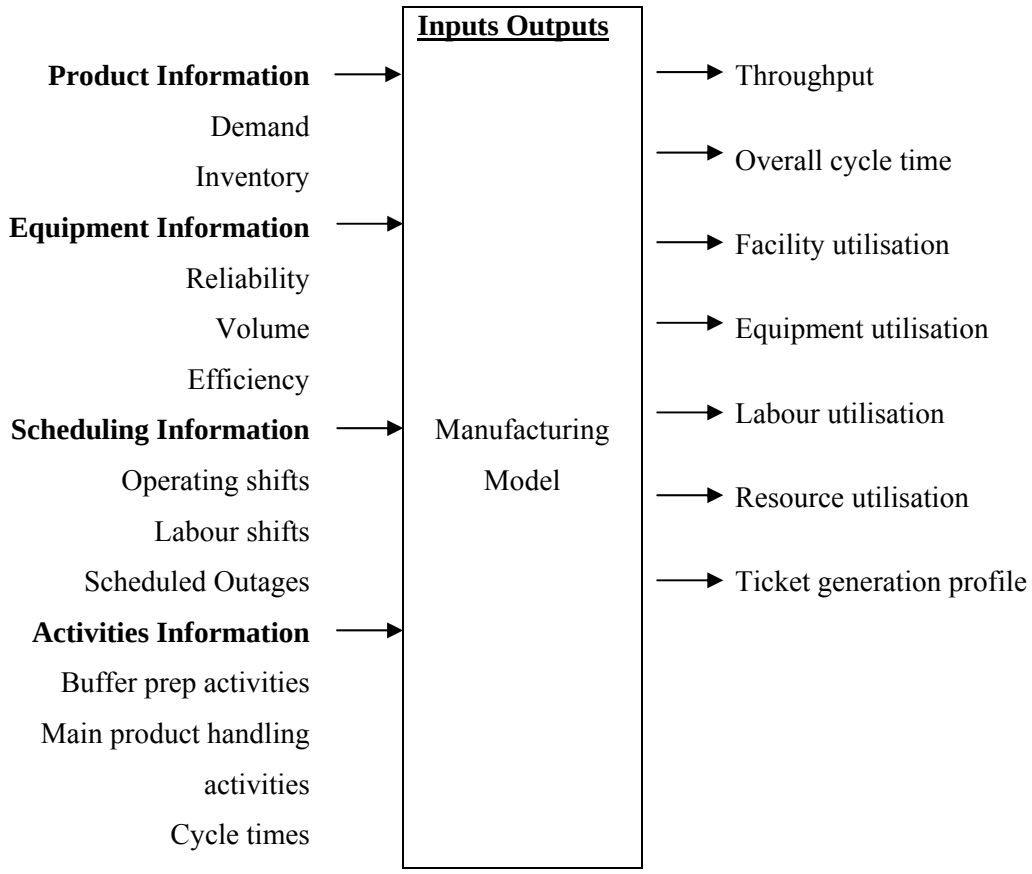


Figure 3.11 Inputs and Outputs of a manufacturing model

3.5.3.1 Questionnaire

It was stated under the proposed methodology that a questionnaire should be used by the modeller to guide the problem structuring phase, allowing them to retrieve the relevant information more efficiently. Table 3.4 is a list of process elements which form the basis of this questionnaire.

3.5.3.2 Templates

All templates library blocks can be found in A.1 and have been built using existing Extend version 7 blocks, in other words, no new blocks were coded in order to maintain a low level of hidden complexity and to allow a modeller ease of understanding.

An important note which must be reiterated here is that, upon review of various different systems and the application of the standards to the various manufacturing ones, while the modelling methodology can be applied to any system, the templates developed here are specific to biotechnology manufacturing capacity management. The reasons for this are simple. Firstly, in order to keep to the 80/20 rule of modelling it was necessary to create templates which were generic only to the degree that up to 80% of the work had already been incorporated into them, thus reducing model development time. Secondly, different systems have so many inherent elements and features that the scope for template development is vast. It was decided that, instead of creating a less comprehensive standard to meet the requirements of all systems, one system would be concentrated on thus maximising usefulness and relevance. Finally, although the templates here have been developed specific to biotechnology manufacturing capacity management, they can be used as a starting framework to take the standard much further, with commonalities across systems being used to adapt many of the template features to cover a far wider range such as Fill/Finish and QC systems. This will be further discussed in the Future Work section.

Table 3.4 Process Elements

Activities	Entities	Resources	Scheduling
Main Product Handling	**Product**	**Labour**	**Shift Patterns**
What are they?	Single or multiple?	Types	Labour set associated with each
Order	Demand	Skills/capabilities	
Parallel?	Available inventory		**Scheduled Outages**
Sub-activities: Pre, run, Post, CTs	Batch size	**Equipment**	- TBF
Synchronisation rules	Batch campaigning? i.e. splitting batch	Number available	- TTR
Priorities	Stability	Failures	
Mutually Exclusive activities?		- Time before failure, TBF	**Production schedule**
Cycles?		- Time to repair, TTR	- Historical?
Labour requirements		Operating shifts	- Random?
Resource requirements		Volume	- Constant?
Clean		Efficiency	
		Flowrates	
Buffer		Area	
What are they?			
Order		**Chemicals**	
Type - simple delay		Amount available	
- sub-activity level		Allocation rules	
Resource/Labour requirements		Costs (if relevant)	
Clean			
Expiry times for buffer makeup		**Utilities**	
Holding requirements		Amount available	
		Allocation rules	
Ancillary		Costs (if relevant)	
Type of clean - simple delay or			
- sub-activity level		**Areas**	
Resources		Conditions	
Expiry		Allocation rules	

3.5.3.3 Data Input

Database template

The database template, as with the library block templates, should require minimised customisation by the modeller and therefore should incorporate all of the commonalities of the systems and models for which it will be used. As stated, the focus of this work has been mainly on biopharmaceutical manufacture and as such, the templates created will cater to these systems. However they can be taken much further and adapted for other systems, building a far more comprehensive library of templates. For example, work flow, fill/finish, QA/QC laboratories and so on. This is discussed further under the Future Work section.

The database template is an Excel file called Input Data (see Appendix A.1) which is used to automatically populate the internal Extend database upon model initialisation. The structure of the tables therefore cannot be changed without these changes also made within the Extend database however with this in mind, the external file has been created in such a way that allows for additional information to be added without any necessary structural changes. For example, the parameter tables for all of the activities have 15 sub-activity rows available, a number chosen based on previously built and analysed models. Furthermore, there are 25 of these activity tables, a number deemed as sufficient for the number of activities usually seen in a biopharmaceutical process chain which is most cases will fall far short of twenty-five.

The tabs in the external database categorise the different data sets required to run a typical biopharmaceutical model. These are listed below:

- Run model – containing the macro to run the model
- General Parameters – with various user defined parameters such as cycle limit for columns
- Activities – a list of all main and makeup activities used a reference point by the database
- Products – this tab contains two very important tables. The first lists the product(s) along with user defined titre(s). The second holds the user defined campaigning schedule which calculates the required number of batches

needed based on the titre and the set demand in kg. The model uses this table to generate items.

- Scheduling – containing shift calculations based on a user defined operating shift and the main facility shutdown which can be used for various scheduling features such as shutting down utilities supply during scheduled shutdown periods.
- Process Areas – containing information on the rooms or process areas in the facility and the shutdown/turnaround procedures surrounding them.
- Equipment information – here the user can define the parameters such as volume, yield, stability and reliability. As with most parameters in the database, the different equipment or activities can have different parameters by varying product.
- Column info –for processes where chromatography takes place and defines the parameters such as column dimensions and dynamic binding capacity.
- SplitCombine – Many calculation surrounding batch cycles are contained here. The primary purpose of this sheet is to calculate ratios in chromatography column cycling based on titre and process yield. However, using a switch in the General Parameters tab, the user can overwrite this functionality and use the tables here to simply split or combine items at different points in the process.
- PSD Data – containing CIP and SIP parameters if relevant to scope
- Utilities – here is a list of the utilities to be modelled and their relevant parameters such as maximum capacity and fill factor. These parameters are useful in line with the flow method of modelling, used in the utilities template which is described in a later section.
- Main/Makeup Activities – these tabs contain the user defined cycle times and labour requirement for all activities, broken down to the sub-activity level.
- Calcs for Lookahead – using the information entered in to the previous sheets, this calculates the run time, and preparation time for each activity and subsequently the estimated start time for each activity based on all those preceding it in the process stream, assuming no constraints. The Lookahead block template uses these calculated values to trigger future events based on what will happen.

3.5.3.4 Data Output

The data output template is an Excel file which is automatically populated with model data once the simulation run has ended. Using data capture blocks placed within each main activity block template, it generates an actual time based analysis of each activity including Gantt charts and utilisation figures. The premise behind this output file is to provide a basic framework for data manipulation with most analysis being based on cycle times. However as the majority of capacity management questions focus largely on scheduling outcomes, it is believed that the output here is sufficient as a starting point for any capacity management model.

3.5.3.5 Extend Template Libraries

Activities Library

The Standards Version 2 resulted in a set of six template blocks dedicated to different activities: Main Primary, Main Secondary, Makeup up, Sub-Activity, Sub-Activity with Failure, Sub-Activity Run. As stated earlier these offered little existing structure and had to be greatly altered in order to meet the requirements of the elements being modelled. As such, the activity block templates have been combined and reclassified and are now as follows: Main activity with cycles, Main activity without cycles, Sub-Activity with failure and Sub-Activity without failure. The rationale behind this new classification is that the blocks required for cycling activities such as chromatography can be incorporated into the main template as default, thus minimising the amount of customisation needed to build these features in, as was done during the mAb case study.

In addition to this library is the CIP/Rinse w/Flow block with can be used to model CIP or rinse only when there is flow from a utilities source, such as the template Utilities block. A separate CIP/Rinse without flow has not been added to the library as one of the existing main activity template structures could be used for that.

Labour Library

There are essentially two ways to model labour. The first is where labour requests are made and processed internal to the process stream. This method is used when the cycle time of the activity making the labour request equals the amount of time that the labour is required. In other words, the labour will be present for the duration of

the activity. Thus the labour is batched with the item before entering the sub-activity or group of sub-activities and then released at the end. This means that all labour blocks are placed internal to the model and are subsequently greater in number.

The second method is used where the duration for which labour is required does not equal the cycle time of the activity. The most common example of this is fermentation, where a growth period may last several weeks however labour will only be needed for a few hours a day. Since labour must be pulled and released during the activity, the processing of the request is externalised. This method uses far fewer executing blocks however, since the labour requests are actual generated items which are thrown to the externalised processing block, the actual number of blocks is actually similar to the first method. Furthermore, since the labour request is processed in a different location to the actual request, debugging is potentially more complex. However, this has been deemed as the best solution to the dilemma of having a labour requirement duration which is not equal to the activity cycle time.

Consequently, there are five labour template blocks: 1) Labour pull (Hrs=CT), 2) Labour release (Hrs = CT), 3) LabourRequest (Hrs<>CT), 4) LabourRequestProcess (Hrs<>CT), and 5) Shift check. The latter block can be used when there is a constraint in place which states that an sub-activity or group of sub-activities can only be started if there is enough time remaining of the current shift.

Resource Library

This library contains three blocks, two of which are based on flow. The first is the Utilities block which is used to model the supply of utilities to the system. The Model Template itself contains this block as default, where it has been externalised placed in the Buffers/Utilities external storage block. However, as the block only accommodates for three utilities or resources, the option of additional ones has been given by placing this block in the library. It works using a very simple concept, linking to the utilities table in the database to retrieve data such as maximum capacity of the supply vessel and fill factor. A sensor is used to control the flow of the utility from supply to facility storage vessel before demand from the system (in the shape of an item entering a pull flow block and requesting flow) pulls the resource. The routing block uses a demand priority system and can flow to more than one place in the system at any one time, allowing the set user parameters and the self scheduling of the model to determine the overall flow in and out of the facility

storage vessel. This allows for a generation rate to be determined based on how fast it has been necessary to supply the facility with the utility, an output parameter which is captured and sent to the output excel file.

The second flow block is the Receive Flow and can be used to either link to the Utilities block, for example to receive steam when modelling SIP, or can be used with any other flow resource, capturing its user defined parameters such as flowrate and fill quantity from the database. One particularly useful instance for this block is where one resource such as an acid used in equipment preparation is made up in one batch however that batch is used in more than one place, i.e. for example three equipment need it at different stages of the process. Realistically, the makeup tank or storage tank is not freed until the entire acid resource has been used by the process. Using items to simulate this hold time can become complex with the use of gates, sensor blocks and database tracking. However using flow allows for the simulation of a tank still holding a certain quantity of resource and only when it becomes empty is the tank released. Therefore the use of flow in this instance reduces the visible, and the hidden complexity of the model and also makes it far more intuitive to the user as they can track the status of resource flow far more easily.

The third block in this library is the Receive Resource block and it is a very basic item receive block which catches a resource and batches it with the primary item in this case, which would be the equipment or batch. This block can be used wherever the rationale applied to the use of the Resource Flow block does not apply, i.e. the resource either does not have associated flow or the modelling of flow is not necessary as the timing of resource allocation is not an issue and can be assumed to be an instance.

Logic Library

This is the most comprehensive of the libraries and contains many of the important function blocks which allow many modelling elements to be modelled in a far quicker and intuitive way. The first block is the most basic one and is the Timer block. Although already placed by default within the Main Activity blocks in the template library it is also present here and can be used to capture any activity time. There are six time elements which can be captured, these are the Pre-Run start and finish, Run start and finish and the Post-Run start and finish. The data can then sent to the output excel file for cycle time analysis.

The second block is the Gate block which is linked to the database and can be used anywhere in the model to control item flow. It simply needs to be linked to the appropriate data tracking table to allow it to know when an item can be allowed through. The conditions surrounding this event are entirely user defined and can be for example, when a batch has finished processing and the next one can be allowed through, or the next campaign must be delayed in entering the system until all turnaround procedures have been completed.

The third block is the Lookahead which uses the before mentioned Calcs for Lookahead Excel calculations sheet to automate the triggering of any activity based on future events. For example, if a buffer is needed at a certain time x, within the process, and it is necessary to follow a 'just-in-time' procedure then that buffer must be prepared based on the current time, the cycle time of all activities between now and x, and the preparation time for the buffer.

The fourth block is the CIP Check which can be used where it is necessary to model CIP expiry based on 'dirty' equipment or limited clean hold time. The block links to the database for user defined parameters such as expiry time and can be used in conjunction with any CIP modelling method, whether externalised or internal, simple or complex.

The fifth block is the Mass Balance block and can be used in models where the scope requires tracking product yield throughout the process. Linked to equipment information defined by the user, it calculates the mass of product throughput based on activity yield and product stability. This block can be placed after each relevant activity enabling the tracking of product throughput across the process without further modeller input.

The final block is the Area Shutdown/Turnaround block which is actually a rather complex structure and can be used for either area shutdown, turnaround or both, where turnaround is the procedures necessary following a product changeover in a multi-product scenario. The block contains many useful functionalities, for example, it can decide whether to synchronise shutdown or turnaround of different process areas or allow a rolling effect (where the procedures begin as soon as the area is ready) or whether to synchronise shutdown and turnaround if they are due to occur within a user defined window, thus reducing area shutdown. This functionality features heavily in Chapter 6 and will therefore be discussed further.

3.6 Conclusion

The standard framework developed in this thesis provides the methodology and modelling tools to allow modellers to construct models which satisfy the six requirement specifications:

- Intuitive to user
- Relevant
- Ease of data input/output
- Short run time
- Maximised reusability and sustainability
- Minimised development time

During the data gathering stages, the framework acts as a guide to improve the efficiency of relevant data retrieval and validation, reducing the construction time by ensuring that a realistic and accurate understanding of the system is first achieved.

The standard templates developed as part of the framework have been designed to speed up the model development process by providing the fundamental building blocks for any biopharmaceutical manufacturing capacity management model. These templates have been designed with a degree of customisation which will allow different process elements to be realistically and more easily captured, but with sufficient generality built in to allow them to be used across different system models.

The development of the standard framework has been an evolutionary process using different biotechnology case studies as a means of evaluating the ability of the standard in aiding the construction of models which meet the proposed requirement specifications. Furthermore, these case studies have been used for analyses such as debottlenecking, dealing with uncertainty and cost analysis, using techniques such as Monte Carlo simulations to test the ability of the standard to create models capable of answering more complex questions. These case studies are described in Chapters 4, 5 and 6.

CHAPTER 4

Application of Standard Framework to a Biotechnology Capacity Management Case I

4.1 Introduction

In biopharmaceutical manufacture there are significantly high costs associated with the running of a production facility, attributable to resources such as labour, utilities, energy and opportunity cost. The latter can be considered an intangible cost and is the cost of using the facility for only one product when it can in theory be used for multiple products. There are therefore two possible reasons for speeding up the production process. The first is to reduce the running costs whilst maintaining production output; if 50kg are produced over a period of 6 months, the cost of resources will be considerably less than if the same amount were produced over a period of 12 months. The second is to free up the facility in order to schedule production of other product(s). This use of the facility in comparison to leaving it idle can reduce further costs such as facility mothballing and labour re-training which can be the result of long shutdown periods.

Speeding up a manufacturing process is not an easy task. It requires the identification of bottlenecks, whether they are resources such as equipment or labour, and the optimisation of activities scheduling such as buffer makeup in time for process demand.

The aim of this thesis section is to illustrate the use of the developed standard framework version 1 in constructing a model capable of being used for this purpose,

looking at capacity management in the large scale production of a biosynthetic therapeutic, henceforth known as BioSynT, in order to maximise facility efficiency.
In section 4.2 a brief description is given of the areas where uncertainty can be found in biopharmaceutical manufacture, particularly in biologics production. Section 4.3 will go on to provide a brief background to the BioSynT case study, describing the model scope and elements. Section 4.4 will discuss the stages of the model construct. In Section 4.5 a deterministic analysis of the model parameters is presented, using process completion time as the objective function. In Section 4.6, a scenario analysis is described, using Monte Carlo analysis to determine possible strategies for process acceleration. Finally, Section 4.7 will evaluate the use of the standard framework version 1 to construct the model.

4.2 Uncertainty in Biopharmaceutical Manufacture

Although manufacturers endeavour to reduce the amount of uncertainty present in their processes, not all elements can be fully controlled even with the most state of the art equipment and automation systems. For example, equipment failure is one element which can be reduced by putting in place regular maintenance checks in order to prevent breakdowns. However one could argue that failure can never truly be eradicated and thus should be taken into consideration when modelling any production process, especially a newly built one.

Also cycle times can vary for many reasons. For example, manual operations such as tray loading/unloading can vary according to labour availability or human error.
In addition to the uncertainties internal to the process, external factors can also play a part. For example, utilities supply across a site can affect the CIP capability of a process if for example, required purified water is made unavailable to it.

4.3 Case study Background

The remit for the BioSynT case study is as follows. The entire production process consists of two streams known as FrontEnd and BackEnd. Both streams have up to now been housed in one facility however a new BackEnd facility was recently commissioned and built, separating the two parts.

The scope of the case study is to determine how fast a certain number of batches (the demand) can be run through this BackEnd facility which contains three major process stages, one of which is the BackEnd process. Figure 4.1 shows these stages where the shaded area containing the three chromatography stages is the Backend process.

A model is required which will generate x batches to represent output from the FrontEnd process which will itself not be modelled. These batches will then run through the model elements, all representing an element of the BackEnd facility or system. The model will then output the 'real' time in which all x batches were processed given the constraints relevant to the real system. For example, a specific constraint is the mode of operation for the process which will be *series*, that is, only one batch will be processed in the BackEnd part of the facility at any one time thus when one batch finishes, the next will begin.

In order to determine how fast the process can run it will be necessary to develop activity and resource utilisation profiles as well as determining the process bottlenecks which may be slowing the process down. Furthermore once these bottlenecks have been determined the next logical step will be to investigate the effect, on the overall cycle time, of removing them i.e. to run scenarios. In order to do this it will be necessary to run the model both under deterministic and stochastic conditions.

Figure 4.2 gives an outline of the steps involved in the determination of these bottlenecks using the deterministic and stochastic approaches and the selection of scenarios based on the outcome of the former. It is based on the assumption that the scenarios chosen are directly linked to the primary bottlenecks. This is believed to be a fair assumption as the selection of these scenarios is based on discussion with a pharmaceutical manufacturer of a biosynthetic therapeutic, modeller experience and the outcomes of the deterministic analysis, all of which put together can give a fair indication of where the bottlenecks may lie. Figure 4.3 shows these steps specific to the BioSynT case study, illustrating the results of the study which will be discussed in further detail in following sections.

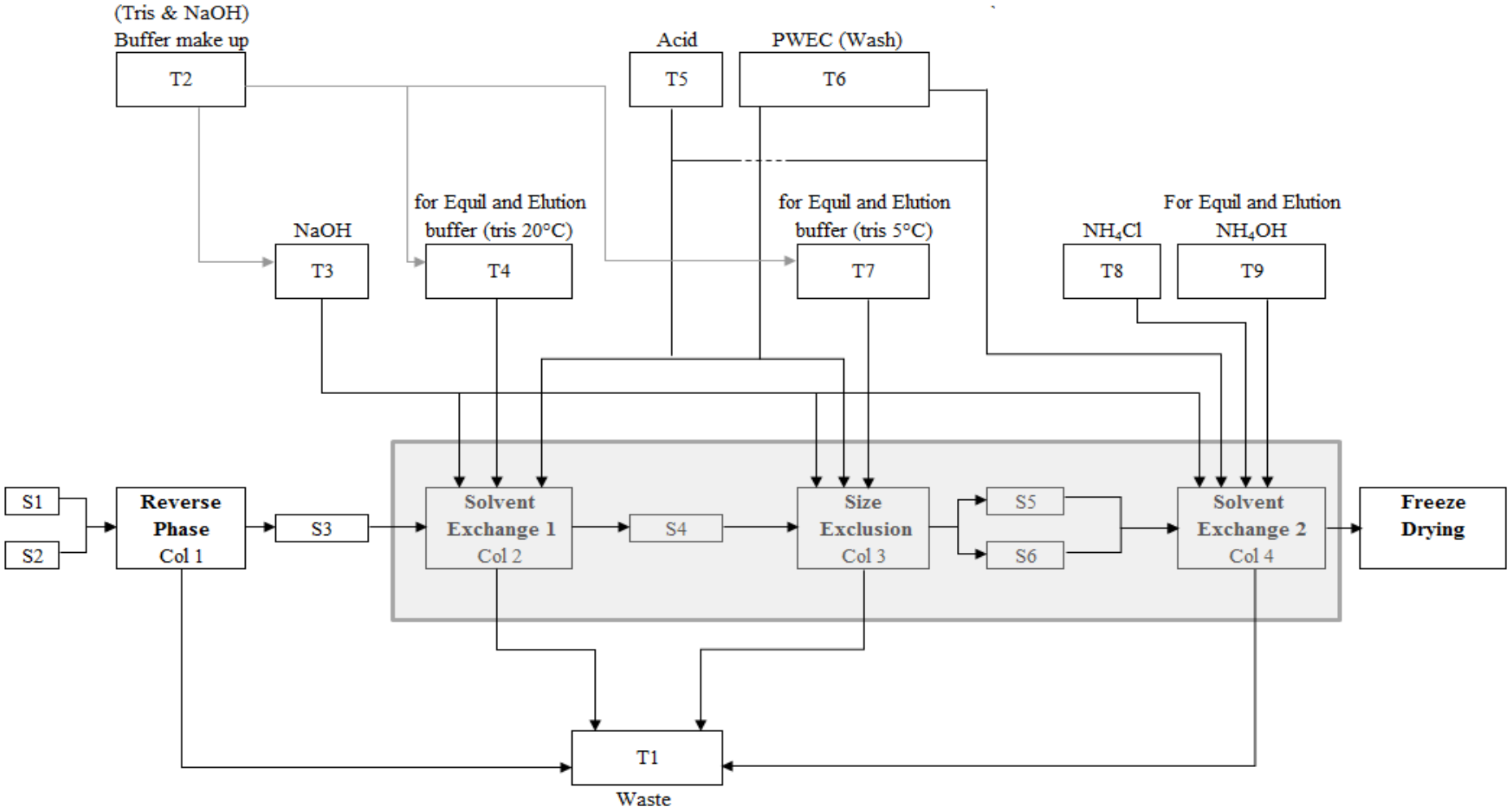


Figure 4.1 Flowchart illustrating process housed in Backend facility including storage vessels (S) and buffer/utilities vessels (T)

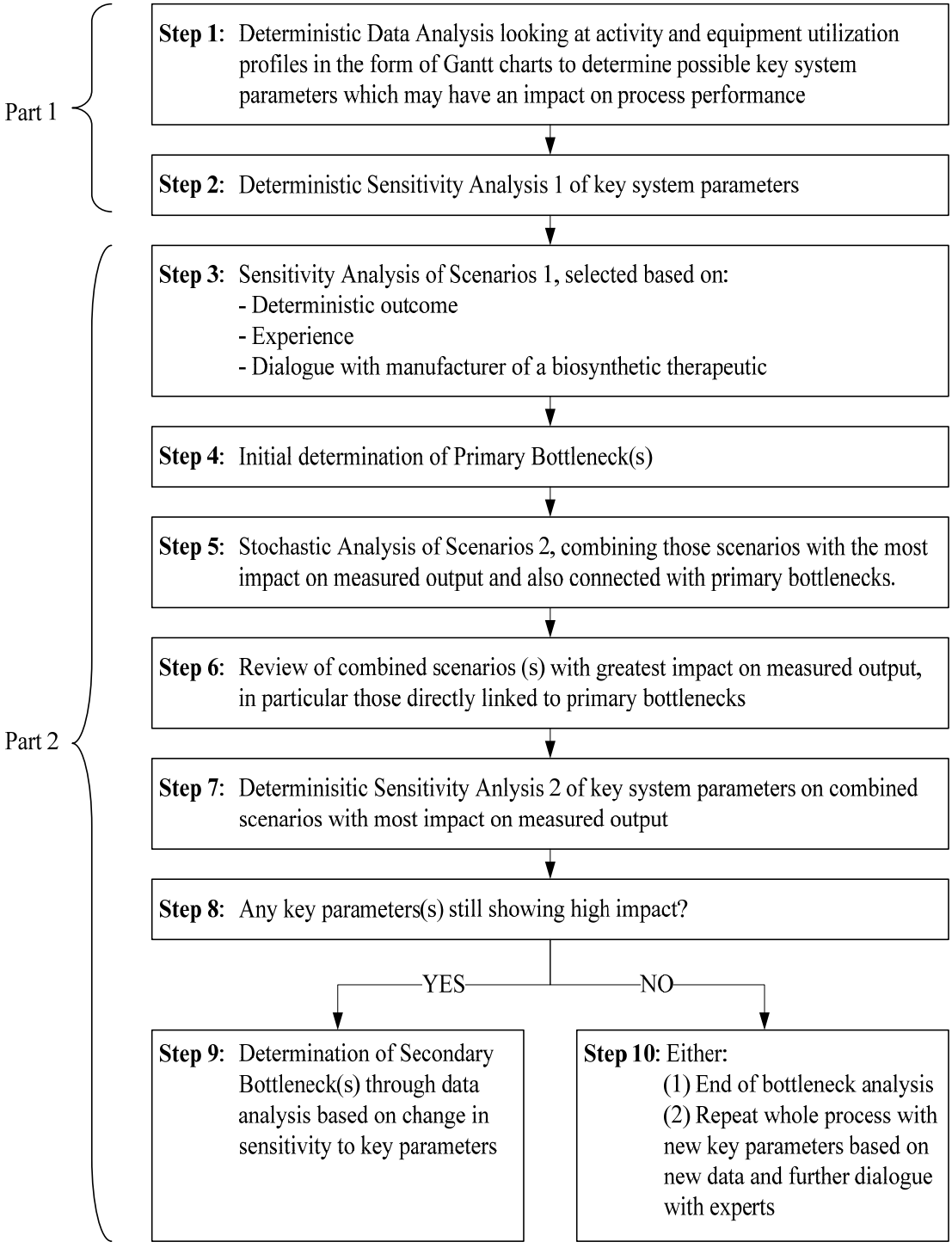


Figure 4.2 Diagram showing the 10 steps involved in determination of process bottlenecks using a deterministic and stochastic analyses

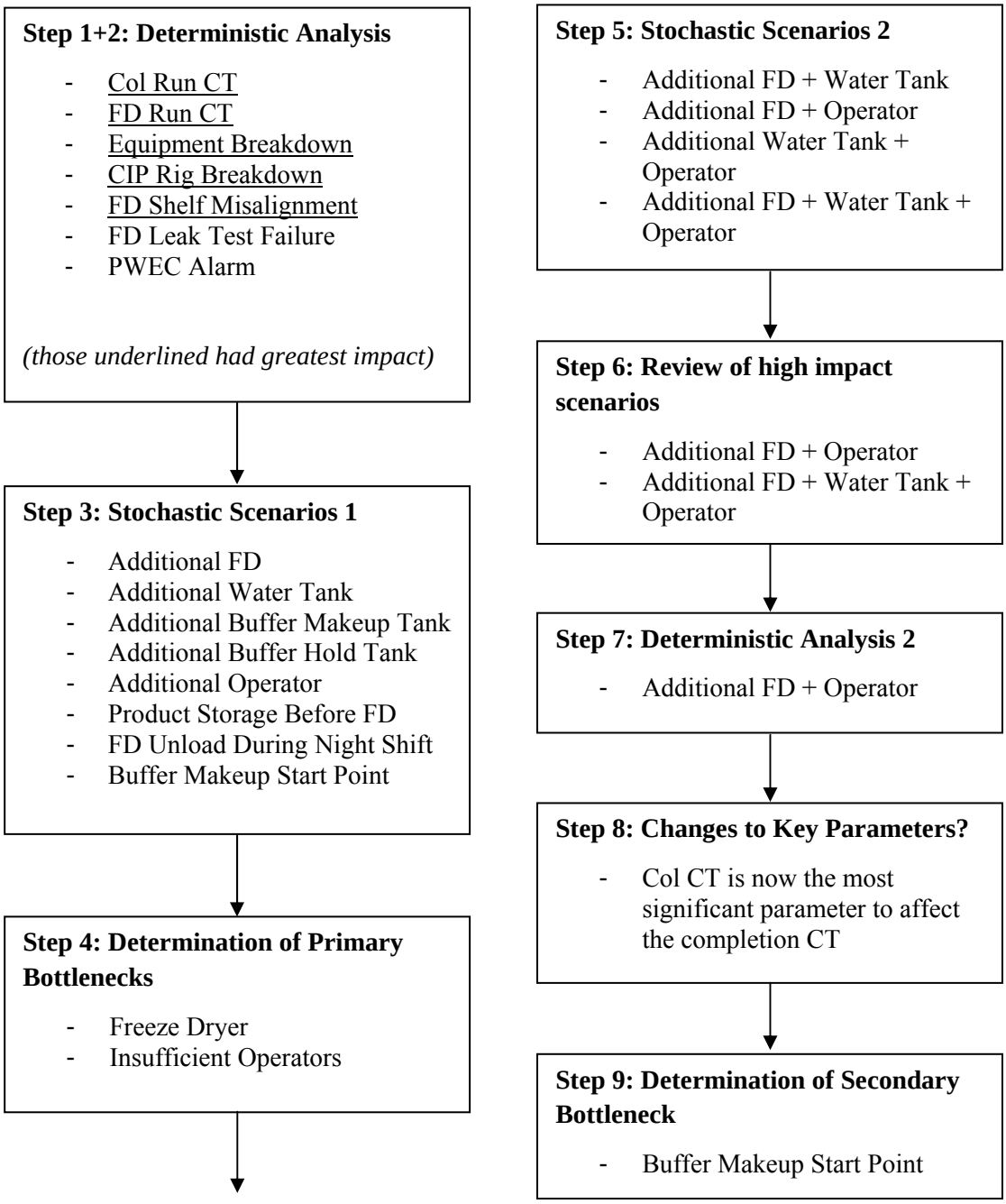


Figure 4.3 Diagram showing steps involved in determination of BioSynT process bottlenecks using deterministic and stochastic analyses

4.4 Method

This section will describe the construction of the BioSynT model using the Standard Framework Version 1, first describing the project scope and process of platform selection and finally the actual constructs seen in the model. Appendix B.1 describes the rationale behind these constructs.

4.4.1 Model Characterisation

Single or Multiple unit operations within problem scope?

In the BackEnd purification process alone there are four chromatography stages and therefore multiple unit operations, in addition to the Reverse Phase step and the Freeze Dryer.

Continuous or Discrete Event?

The primary items which enter the BackEnd purification stream are containers of frozen intermediate from the FrontEnd purification process. Each container enters the first step individually as a batch. Each batch is processed at each unit operation, pooled and then sent to the next unit. Law and Kelton describe discrete event modelling as the 'modelling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time'. Events, defined as 'instantaneous occurrences that may change the state of the system, occur at these points in time'.

The analysis of resource usage according to batch 'generation' in the BioSynT process is a discrete event system and not a continuous one.

Dynamic simulation

The system consists of a series of both sequential and parallel steps, with batches moving from one step to the other. Each step also has a cycle time associated with it. Therefore the time dimension must be modelled.

System Elements which must be Modelled?

The case study will examine the scheduling of the BackEnd BioSynT process, taking into consideration the effects of the various system constraints and rules, such as

labour allocation, on the schedule. Therefore the system elements which must be captured are the, activities (both main and ancillary), resources (labour, equipment. buffers) and entities (batches). The numbers/amounts of the resources along with the time attributes of the activities will require tracking in order to ascertain the effects on the schedule.

Metrics

Since the remit is to find out how fast the required number of batches can be processed, the two main metrics of the model should be the number of batches gone through and the time taken to process or the overall completion cycle time. Also there are certain constraints on the system, mainly in the form of resource availability, the effects of which should be measured. Therefore resource utilisation should be an output of the model, with visual aids such as Gantt charts and single figures such as percentage utilisation to give a clear indication of resource usage over time as well as overall. Finally, one element of the system, of interest to the BioSynT client, is the rate of ticket generation during the running of the process. These tickets are documents which are generated when an activity starts and closed when the activity has ended and the responsible labour has signed off on it. The number and location of ticket generation must therefore also be tracked.

For the purposes of this case study, due to the fact that it is more of a scheduling problem, it will not be necessary to model the biochemistry across the purification steps, therefore mass balancing will not be required.

Constraints

The constraints on the system are mainly resource availability and time constraints in terms of limits for product holding in storage. A list of the given constraints under the scope can be found in Table 4.1.

Table 4.1 Summary of BioSynT case study scope

Single or multiple activities / unit operations?	Multiple
Continuous or discrete?	Discrete
Static or Dynamic?	Dynamic
System elements	• Entities: - Batches • Activities: - Main activities e.g. chromatography - Sub-activities e.g. equilibrate - Ancillary e.g. CIP • Resources: - Buffers - Equipment - Labour
Metrics	• Throughput • Cycle times • Resource Utilisation • Ticket generation
Constraints	• Series mode of operation • Labour allocation for sub-activities • Synchronisation of equipment preparation with resource availability • Storage hold time limit • Labour availability • Share buffer makeup and hold tanks

4.4.2 Platform Selection

In order to meet the requirement specifications of a simulation model it is important to choose the right platform to construct the model on. There are various types of simulation software on the market; the following will describe the main platform types, specific to examples given for each platform type.

There are four major types of simulation platform which can be used for simulation of such systems. These are Spreadsheets (Excel), Mass balance (Batch+), Discrete Event (Extend) and Equation Oriented Modelling Packages (MathCAD). In order to choose a suitable platform for this particular model construct it is necessary to perform an analysis of the capacity management problem.

Spreadsheets such as Excel are suited to less complex systems where modelling of the dynamic evolution of system objects is not required. In effect spreadsheets are good for creating snapshots of the system at certain points in time, where simple (relatively) calculations are sufficient for system analysis.

Engineering calculation or mathematical modelling platforms such as MathCAD are similar to spreadsheets in that they are better suited to less complex systems where dynamic evolution is not required. They allow for stochastic modelling with tools such as Monte Carlo. They are suited to the creating and documenting of calculations and can be used for such things as mass balancing and chemical calculations.

Mass Balancing platforms such as Batch Plus or SuperPro Designer are suited to detailed analysis of the chemistry behind unit operations, allowing for analysis of the component makeup as a result of reactions/binding and so on. These packages have graphical user interfaces (GUIs) which help to represent the system.

Discrete Event simulation platforms such as Extend are suited to more complex, discrete, dynamic systems where detailed mass balancing is not required, but rather a more overall system analysis, with entities, resources and control systems; for example, looking at scheduling and resource utilisation. These also have GUIs.

Table 4.2 compares the four major types of the simulation tool, against the requirements of the BioSynT case study as defined under the project scope. The comparison has been carried out using a scoring system where relevant, representing the extent to which each platform is able to model each requirement.

Ultimately, as the system being modelled is a dynamic, discrete event one, this eliminates spreadsheets and engineering/mathematical platforms.

Table 4.2 Showing the different types of modelling platforms and their modelling capabilities in terms of model scope requirements. Scores given out of 5, 0=not at all, 5=fully meets requirement

	Spreadsheets (Excel)	**Mass Balancing** (Batch Plus)	**Discrete Event** (Extend)	**Equation Oriented** (MathCAD)
Single or Multiple Activities	Single (too complex at multi)	Multiple	Multiple	Single
Continuous or Discrete	-	Discrete capabilities (see below)	Discrete with continuous capabilities	-
Static or Dynamic?	Static	Static (Aspen Plus Dynamics can extend static models to dynamic)	Dynamic	Static
Stochastic capability?	3	4	5	3
Mass balancing?	- statically	5	- basic	- statically
System Elements	3 Not dynamically	4	5	2 Not dynamically
Metrics	0 Outputs required are based on dynamic calculations	3 Better for platform defined metrics e.g. biochemistry	5	0 Outputs required are based on dynamic calculations
Constraints	1	3	5	1

This leaves mass balancing and discrete event tools. Since the scope does not require mass balancing, rather more user defined outputs, Batch Plus would not be a good candidate in this case.

It is believed that Extend would be more suitable as it meets all of the requirements.

4.4.3 Model Construct

SYSTEM ELEMENTS (REAL PROCESS FEATURES)

Activities (Product Handling)

These are the four chromatography stages followed by the freeze drying step. The intermediate storage steps are also considered product handling. Hierarchical blocks represent all these functions. As the project scope requires a more detailed analysis of resource utilisation and constraint modelling, it was decided that the sub-activities of each main product handling activity would also be modelled i.e. the preparation of the equipment, the running/processing of the product, the post run cleaning of the equipment. Each of these sub-activities is a further hierarchical block containing Extend blocks which represent processing time. Modelling the further level of detail ensures that the model remains within the scope by answering the questions of scheduling and utilisation more fully and therefore contributes to its reliance and reusability.

Activities (Ancillary)

The only ancillary activity is CIP of the main product handling equipment. This activity has been externalised and sits on the top most level of the model. The externalisation of this activity means that equipment are 'thrown' here for their CIP processing rather than each containing their own CIP block. This reduces the visible complexity of the model by using far fewer blocks and concentrating this particular function in only one place. Also this block can be used to represent CIP for the other products under multi-product manufacture as it sits external to all process blocks.

Activities (Preparation of Buffers)

There are two types of buffer preparation related activities. The first is the actual preparation of all buffer materials. The second is holding of some buffers before use within the process. Like the Ancillary activities, these have been externalised.

Within each of the buffer blocks there are sub-activities representing the makeup of that buffer, again, to ensure enough level of detail to capture the correct utilisation of the equipment and consumables used in making the buffers i.e. water for cleaning the tanks.

Entities

The entities in the model are the primary items or batches. As inventory exists within the model some batches are held after the Reverse Phase in frozen state, and some before the Reverse Phase. The batches are numbered sequentially in order to track their progress and the process data associated with the processing of each, for example, cycle time through the freeze dryer.

Resources

Labour has been modelled in resource pools which are kept externally on the top most level of the model. Labour allocation is based on whichever activity block executes a request first.

The equipment are held in 'pallets' which are in effect holding blocks where the labour sits. Gates control whether each equipment leaves its holding block to be prepared for processing.

The consumable resources in this system are the buffers. In Appendix B.1.it is stated that these should be modelled in resource pools. However in this case there is a third method which must be considered to account for consumable resources which are used over a period of time when the equipment where they are dispensed from is held up during that period. This method is the use of Flow; for example one lot of acid is prepared once and used for three different washes in three different consecutive main activities. The tank in which the acid is prepared cannot be cleaned and used for the next lot until all required material has been drawn from it with the remainder sent to waste. In order to simulate the holding up of this tank, flow must be used because a) if the acid material is held in a resource pool then this externalises the contents of the vessel and makes it difficult to track its contents and b) if the acid material is unbatched and represented as an item then this creates a tracking nightmare. Flow internalises the resource, representing more truly the contents of the vessel. It also allows better control as the vessel is actually held up until the required contents have flowed out

LAYOUT

Physical Layout and Decomposition

Discrete event simulation modelling in Extend involves hierarchal levels containing blocks. Functional Decomposition has been used and therefore the hierarchical blocks represent the activities in the real system. The top level of the model looks like that in Figure 4.4 with all 'storage' blocks (blocks containing externalised elements, global arrays and resource pools) positioned at the very top of the workspace.

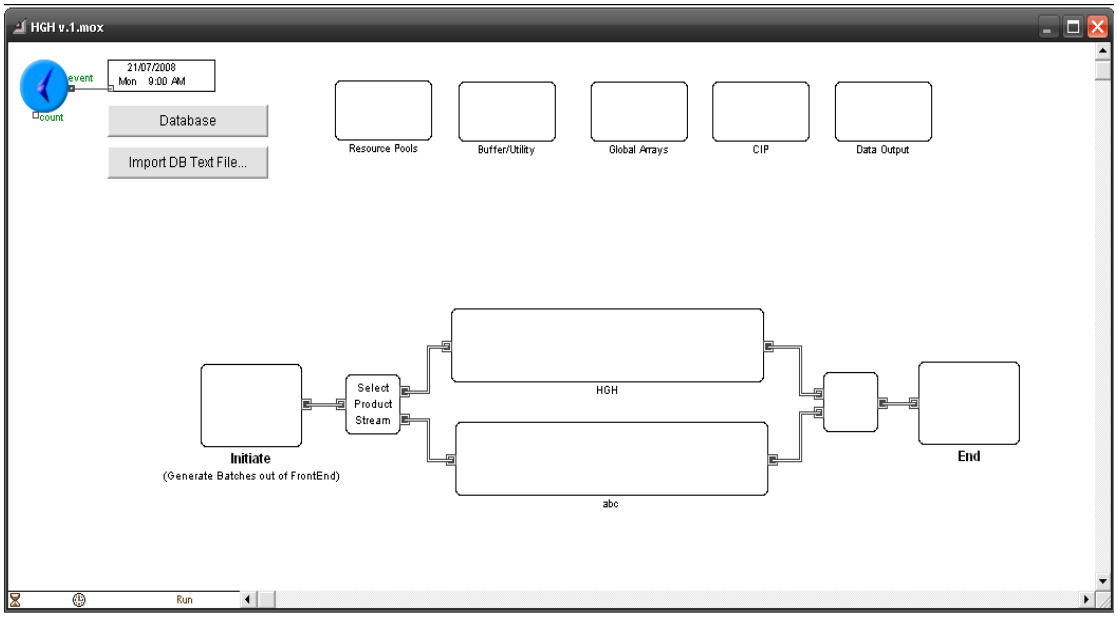


Figure 4.4 Top most level structure of the BioSynT model

Block Construct

In addition to the higher level structures a standard structure has also been devised for the sub-activities and blocks below the main activity level. As stated under the Standard Framework, in situations where there are equipment (or indeed any entity) which have their own activities independent of the primary item, the following structure for the arrangement of sub-activities can be used.

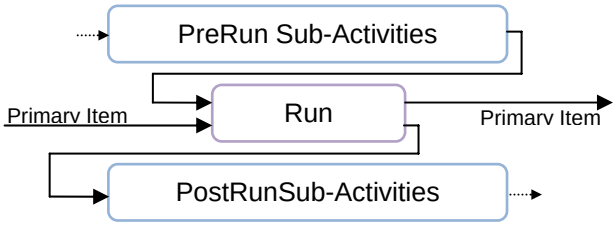


Figure 4.5 Diagram showing the sub-activity structure within the BioSynT model

Note that this structure can be used in part, that is, there may not necessarily be pre-run or post-run activities before or after the primary item comes in.

Item Transfer

Item transfer occurs using the conventional IDEF0 block and connector method to create an intuitive model. The model also flows left to right within all hierarchical blocks. This is due to both model clarity and also the fact that Extend executes blocks in a left to right direction.

Item Flow Control

There are certain areas of item flow which must be controlled. This is done using a gating system whereby a series of gates are used to synchronise activities by doing checks such as 'are all required buffers ready?' and to control item flow based on equipment status. In order to create more generic blocks, the basic structures of these gates are identical; they only differ in the data arrays which they reference.

MODEL INITIATION

Primary Items Generated

The main items generated in the model are batches. The required number is generated all at once and their path through the model is scheduled by the gating system.

MODEL LOGIC

Metrics

Cycle Times: Although cycle times of individual activities are set in the database, the actual process times vary due to the constraints and delays inherent to the running of the process. A simple block has been used to record the time at various points in the

model. These times are stored under the appropriate attributes e.g. BatchStartTime and later sent to Excel where a Gantt chart is automatically generated.

Throughput: This is quite simply the output of the model. The number of primary items which leaves the model is the number of batches that have been processed.

Equipment Utilisation: As all cycle times for all activities are sent to Excel it is quite easy to calculate the equipment utilisation (Eq.Ut) using the following equation:

$$\text{Eq.Ut} = \frac{\text{Total Actual Processing Time}}{\text{Time Finished Processing Last Batch -Time Started Processing First Batch}} \times 100\% \quad (4.1)$$

Total Actual Processing Time is the time that the equipment spent processing the batch minus all the gaps due to waiting for resources. This time however does include the downtime.

Resource Utilisation (Labour): It is useful to track the usage of labour in order to determine exactly what number is required to run the process or to analyse the effects of varying availability.When constructing the model only two rules or constraints regarding the labour allocation existed, 1) each subtask must be started and completed by the same operator and 2) the unloading of the freeze dryer must be done during the day. The first is rather logical since every task must be signed off with most lasting only an hour, some only 10 minutes.

In order to model the first rule it was initially considered that a logic be placed before every sub-activity in order to check that it could be completed during that shift. However there are in total 42 of these sub-activities which would mean a rather significant increase in visible and hidden complexity. Therefore it was decided that each main product handling activity be broken down into ‘sections’ of sub-activities which if started must be finished but which can have time gaps in between without ‘expiry’. For example a column could be prepared and equilibrated and then wait for the next ‘section’ which would be the processing of the batch. Note that since each shift lasts 12 hours, this would be the maximum (and unlikely) time gap. It was

assumed that this would be allowed, for example, equilibration could be done and left without 'expiring' until the next shift.

Each main activity was broken down into the following chunks or sections

- RP All sub-activities
- Solvent Exchange 1 PreRun, Run, PostRun
- Size Exclusion PreRun, Run, PostRun
- Solvent Exchange 1 PreRun, Run, PostRun
- FD PreRun, Load, Unload, PostRun (no labour for run)

At the beginning of each section where a worker is needed a shift rule block is placed which checks the following:

- What is the current time and therefore current shift, night or day
- How far into the shift is current time
- What is the cycle time of the activity
- Assuming that the section can begin as long as the current time is greater than 1 hour before the end of the shift, an operator is pulled from the resource pool. However if it is in the last hour of the current shift, a delay is forced which waits until the end of the shift before an operator is pulled.

For the freeze dryer the same block is used however only day shift worker is made available

Ticket Generation: During the construction it was decided that the generation of work tickets should be tracked in order to see how long each is kept open before being signed off and more importantly how many are open at any one time. In order to tackle this, a 'ticket' block was created. This block is used for the opening and closing of a ticket. When a ticket is opened (for example when a column is going to be packed) the block is turned into an opening ticket place entering a 0 in the top left hand corner. The ticket number is then entered at the bottom. This number is set and can be referenced in the global arrays block. The block will then automatically write to the global array when that ticket was opened. Subsequently the contents of the global array are sent to Excel at the end of the simulation where a Gantt chart is created showing ticket generation.

Shift

The facility runs on a 24hour basis with a current 12hour labour shift. There are two significant rules based on the labour shift which must be considered:

- No activity can be started by a worker towards the end of a shift i.e. a worker must be able to start and finish an activity
- Unloading of the freeze dryer can only occur during day shift

DATA

Data Transfer

Data generated by the model is transferred to Excel for analysis from either global arrays or attributes. Current data is all time related and therefore the output charts are all Gantt.

Database

All data is entered into the excel database and transferred to Extend via the SDI link.

4.4.4 Case Study

The next section will focus on the results of the case study, looking at the deterministic and stochastic analyses carried out. This section will briefly discuss the key base assumptions used to form the basis of the study.

4.4.4.1 Key Base Assumptions

The model was run with the following parameters and assumptions. Those made variable have been marked as such. All others were hardwired into the model structure.

General

- Number of runs = 1
- The process is only run in series mode, that is, only one batch can be in the BackEnd process at any one time however at the same time there may be one processing in Reverse Phase and also Freeze Drying.

- The model time horizon is 1 year, based on a 310day year, 30day month. This is based on the assumption that the overall process will run at 85% efficiency
- Shift: 12hour labour shift, 24hour process operating shift (**Variable**)
- Assuming 40kg per annum required throughput, this equals 40 batches required; 5 held in frozen inventory after RP and 35 before RP (**Variable**)
- Assuming that product will remain viable in intermediate storage for 24hrs and 48hrs depending on stage
- The Reverse Phase operation will only run during running of the Freeze Dryer
- No equipment is prepared for processing a batch until a check has been carried out to ensure all required buffers are available
- The Freeze Dryer requires up to 4 people to load and unload. It has been assumed that 2 will need to be pulled from the regular shift and that the remainder will be available as and when and are therefore not a constraint (**Variable**)
- Columns can be equilibrated and left for up to 12 hours
- There is no storage between the last column, Col 4, and the Freeze Dryer. The batch must be eluted straight to the dryer. Therefore product is kept in S5 and S6 but only for up to 24 hrs. Col 4 is then run just in time to elute to FD.
- Reverse phase does not go to CIP. It is regenerated and re-equilibrated straight after elution and put into storage

Failures

- Buffer tank probes fail 1/10 or 1/20 (depending on tank) runs/make-ups (**Variable**)
- PWEC tank supply stops 4 times in a 12hr shift for 30 minutes (**Variable**)
- CIP rig breaks down once per 24hr operating shift for 6 hours (**Variable**)
- FD leak test fails 1 out of every 3 runs. It takes 2 hours to investigate before repeating the test (**Variable**)
- FD shelf misalignment occurs 1 out of every 4 runs. It then takes 8 hours to unload and fix before reloading. (**Variable**)
- There is a breakdown in the Main Product Handling equipment (columns and FD) once a week with a different one every week. Since there are 5 pieces of equipment, time-before-failure for each will be five weeks. (**Variable**)

- Assuming that if equipment breaks down while processing product, that batch or cycle lot will not be discarded.

Buffers

- Buffers A and B together take 5 days to make up (20 batches worth)
- The order of buffer makeup is fixed i.e.:
  Tris 20 → Caustic 100 →Tris 5 → Caustic 10 → Caustic 100 → Buffer A→ Buffer B
- Buffers for RP (A and B) are made up once the level of buffer remaining reached below 5 batches (**Variable**)
- Buffer makeup for each batch begins only when the final column, Col 4 has been cleaned and placed in storage

Labour

- 2 operators do everything. PT only helps with load and unload of FD (**Variable**)
- The 2 operators can work simultaneously on all buffer makeup. They can do this during process operation spending a percentage of their time on each task
- There is no unloading of the FD during the night shift

Concerning Scenarios

- When there are two T415 tanks (used for caustic, tris and RP buffers) the tanks do not make the same buffers, rather they share the burden. In other words one tank will only make caustic, the other will be dedicated to Tris and RP buffers.

4.5 Results and Discussion

4.5.1 Initial deterministic analysis

In order to perform a worthwhile scenario analysis it is first necessary to determine which parameters will have the greatest impact on the measured output or process performance. This is where a deterministic sensitivity analysis must be used. However, when modelling a complex system there are a myriad of parameters which could be tested and it is important to be able to eliminate all but only those with potential impact in order to save time. The following section will discuss the setting

up of the deterministic case, including the selection of parameters for the study and the results of the analysis.

4.5.1.1 Setting up the deterministic case

In order to carry out a worthwhile deterministic analysis it is necessary to first ascertain which factors have the greatest impact so that only those may be looked at. The following examines the possible factors or parameters.

Throughput

Firstly based on the assumptions outlined in section 4.4.4.1 the throughput was calculated to be 100% i.e. 40 batches output. This is due to the fact that the allocated time horizon of 310 days is more time than required to process the required throughput considering that each new batch goes through on average every 6.46 days, requiring a total of only 258 days for all 40 batches. However this cycle time which equates to 8.62 months is still rather high if the desired outcome is to reduce production time to 6 months or below, in order to shut down the facility for half the year or more importantly, use if for another product. Therefore it would be interesting to deterministically investigate possible ways in which this cycle time could be reduced. One obvious way would be to reduce the cycle times of the main product handling materials.

Utilisation

Three types of utilisation were tracked: equipment, facility and labour. Table 4.3 shows the results for the base case. In order to show how these results differ for different scenarios later incorporated, the utilisation figures for the two main bottlenecks have also been included. The third column shows that when an additional operator is available taking the number of operators to 3, the labour utilisation drops from 70% to 47% which is expected as the burden of labour requirement is spread across a larger number of operators. This column also shows that the utilisation of the freeze dryer increases to 93% which is again, expected as later work shows that the FD is the second bottleneck in the process behind labour availability. The fourth column in the table shows that when both the bottlenecks are removed, the utilisation figures for the labour and freeze dryers decrease.

Table 4.3 Utilisation figures for base case scenario, addition of an operator and the addition of an operator plus a freeze dryer

	Base	**3 Operators**	**3Ops + FD**
Facility	84%	69%	64%
RP (combined)	52%	84%	95%
G15	19%	23%	24%
Sephadex	26%	32%	35%
G25	42%	52%	56%
FD 1	78%	93%	51%
FD 2	-	-	50%
Labour	70%	47%	43%

One strange feature of the results is that the utilisation figures for the equipment increase as the bottlenecks are removed, particularly the labour bottleneck. The reason for this can be found by looking at equation 4.1.

Using the first solvent exchange column, Col 2 as an example, when the elements of the equation are compared with the base case scenario and the addition of an operator the following results are seen:

Table 4.4 Elements of equation 4.1 calculated for base case and 3 operators for Col 2

	Base case	**3 Operators**	**% Change**
Total Actual Process Time	47.5	46.9	-1.3%
Time Started	0.24	0.24	0.0
Time Finished	253.2	205.5	-18.8%

When plugged into the equation the result is a greater value for equipment utilisation with application of the labour scenario:

$$\text{Equipment Utilisation} = \frac{\text{-1.3\%}}{\text{-18.8\%}} \text{ x } 100\% \qquad (4.2)$$

In terms of what is actually happening in the model, when additional labour is available the times during which the equipment sits waiting for labour is reduced. This reduces the total actual processing time for the equipment (as this includes all the gaps due to labour unavailability). But this effect is local to Col 2 hence the relatively small decrease of 1.3%. However throughout the process the increase in labour availability also has an impact on every other equipment, meaning that batches can be processed faster. This global impact is what affects the 'time finished' element of the equation, hence the large decrease of 18.8%. Therefore although the overall impact of labour availability may be an overall reduction in cycle time, locally the equipment utilisation figures will increase due to the use of the above equation and the assumption incorporated. This also explains why the overall facility utilisation decreases with the removal of the bottlenecks, particularly labour.

Cycle Times

The cycle time of each batch through the BackEnd alone varies, averaging 4.6 days, with the overall cycle time for all batches through RP–BackEnd–FD being 258 days.

Figure 4.6 shows all main product handling activities from Reverse Phase to Freeze Drying including all intermediate storage, for the production of one batch. Firstly, Reverse Phase is not the first activity to take place as there is inventory product held in freezers between RP and Backend. This means that the latter can begin processing immediately taking defrosted material from pool tank S3. From here it enters the first solvent exchange and then subsequently gets eluted into pool S4. From here a significant overlap can be seen within the BackEnd process due to the use of pooling vessels S5 and S6. Here the batch entering size exclusion is split into 6 lots where each is cycled through the column. As a batch is eluted it enters straight into a storage vessel. The first half of the 6 lots fill storage vessel S6 while the latter half fill vessel S5. This is the first overlap. The second occurs when, while storage vessel

S5 is being loaded with material from the size exclusion column, vessel S6 begins to unload its contents into the second solvent exchange. Finally the freeze drying stage begins on the 4th day. Therefore the BackEnd section of the process takes approximately 4 days.

What the Gantt chart shows is that the pre- and post-run activities are not the factors with high impact on the overall process cycle time as these are done well in advance in the majority of cases, ruling them out as bottlenecks. It is the cycle times for actual product processing which should be investigated particularly in the case of the freeze dryer which takes 55 hours to run only one batch.
Furthermore the fact that the pre- and post-run activities are not bottlenecks suggests that the buffer supply is adequate and not in itself a bottleneck. Therefore buffer supply will not be looked at under the deterministic sensitivity analysis.

The vertical axis corresponds to different main product handling activities while the horizontal bars correspond to different sections of activities within the main product handling ones. These sections are PreRun, Run and PostRun. The darker shaded bars along each horizontal grid line represent the processing of batch material and the lighter shaded ones represent the pre-run and post-run activities associated with the equipment.

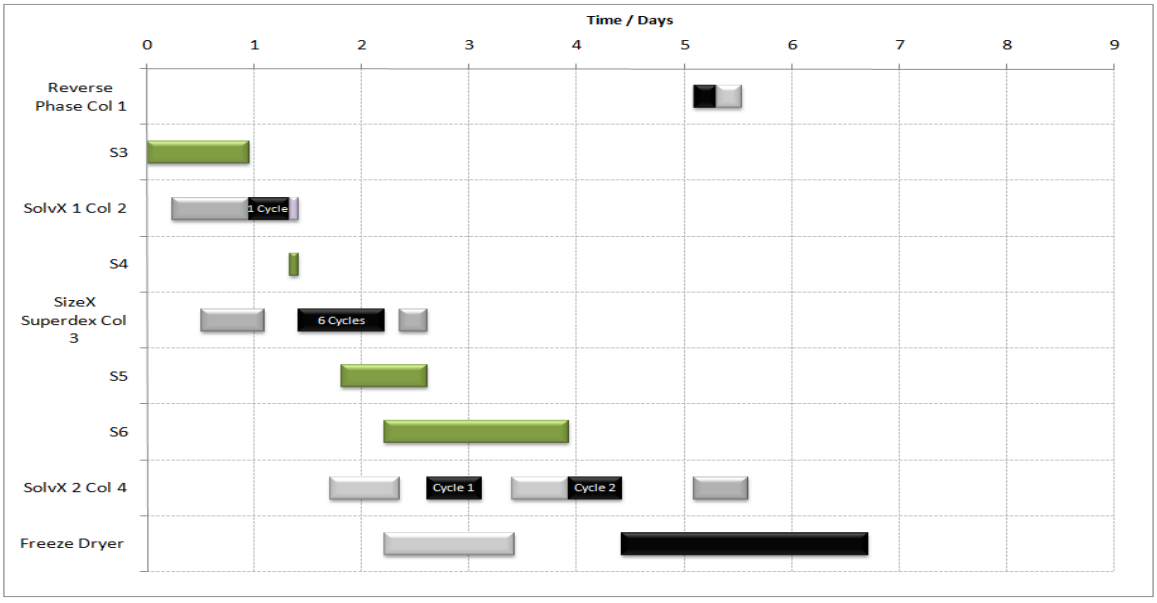


Figure 4.6 Gantt chart showing the Backend facility process for 1 batch. Running conditions: (1) 2 operators available, (2) 40 batch demand - 5 of which come from frozen inventory post RP.

Labour

The labour used for each section is tracked using attributes. The top chart in Figure 4.7 shows labour usage for a 31 day period. In order to see the effects of labour availability on the process it is useful to compare this chart with the process Gantt. The bottom chart in Figure 4.7 shows the main product handling activities for the same 31 day period.

The labour chart shows that the labour usage fluctuates constantly with not all available labour being utilised all the time. This is unsurprising as the labour utilisation figure is 83%.

Figure 4.7 Labour usage and main product handling activities for the same 30 day period

Table 4.5 Showing days during which labour is fully utilised and therefore unavailable along with times during which process sub-activities wait for labour availability

	Range of days during which labour is unavailable											
	0.5 - 3.4	4.4 - 8.5	8.6 - 9.6	9.6 - 10	11.8 - 16.9	17.1 - 17.3	17.6 - 19	19.1 - 20.7	21.7 - 22.4	22.5 - 23.8	24 - 25.7	26 - 31.4
Times where gaps in process	1.1	4.5	9.3		11.2		19	19.1	22.1	22.8	25.7	26
	2.2	7.3			11.8				22.3	23.1		29.5
	3.1	8.2			14.8							30.2
	3.2				16.9							
	3.4											

However although the utilisation may not be 100% the scheduling of events means that labour is required when unavailable; Table 4.5 shows more clearly the days during which labour availability is zero along with the corresponding gaps during the main production process when activities are waiting for labour to become available. It shows that all 23 gaps fall during the times when labour is being utilised elsewhere. Therefore labour is a factor which affects the overall cycle time. However, it will not be included under the deterministic analysis as a potential uncertainty but will rather be set as a scenario. The reason for this is that the process simply will not work with less than 2 available workers and therefore a distribution for uncertainty would be heavily biased towards the positive.

4.5.1.2 Deterministic analysis parameters

Below are those parameters to be included in the deterministic study.

Equipment Breakdown Frequency and Time to Repair: The frequency of breakdowns in the process equipment is on average once per week with repair time being, again, on average 6 hours. It seems that both these numbers fluctuate, particularly due to the fact that the process is new and that the modular nature of the facility has meant more difficulty in gaining access to areas and instruments for repairs.

CIP Rig Breakdown: According to the process team, the new CIP rig is rather temperamental and can break down with high frequency. Unlike the equipment

breakdown however, only the breakdown frequency will be changed, keeping the time to repair constant.

Freeze Dryer Manual Operation Cycle Time: The Loading, Sampling and Unloading of the freeze dryer are carried out manually and take several hours. Factors such as illness, labour shortage or time of day can contribute to slower labour input. On the other hand, extra labour, or good moods could mean an increase in efficiency.

Column Cycle Times: Changes in product potency can lead to changes in the charge volume resulting in different cycle times. These would typically be in the region of plus/minus 10%. Furthermore, variations in flow rate can also affect this.

FD Leak Test Failure: The failure of the freeze dryer leak test is a common occurrence and can cause several hours delay for investigation and repair. The test must then be repeated.

FD Shelf Misalignment Frequency: Shelf misalignment occurs during loading of the freeze dryer and is a common occurrence. The delay caused is significant as the FD must be unloaded and reloaded before continuing.

PWEC alarm: the alarm on the cold PWEC supply tank sounds rather frequently during every shift. Since the makeup of all buffers relies on PWEC supply, 30 minute stoppages may have an effect on the speed of availability of the buffer. However since buffer supply is unlikely to be a bottleneck, it is believed that this will be one of the lowest impacting factors.

4.5.1.3 Deterministic analysis results

Figure 4.8 is a tornado diagram showing the parameters tested and the impact on the process completion time for all 40 batches. It shows that the biggest impact is seen when the frequency of the freeze dryer shelf misalignment is changed, particularly when it is increased. It is this negative impact on the overall cycle time which puts this parameter higher on the diagram than the chromatography column cycle times which show a bigger range of percentage change in comparison. There is a similar pattern observed with the equipment breakdown and freeze dryer cycle time. The lowest impacting parameters are the FD leak test failure frequency and the PWEC

alarm frequency. These will consequently not be considered in the stochastic analysis.

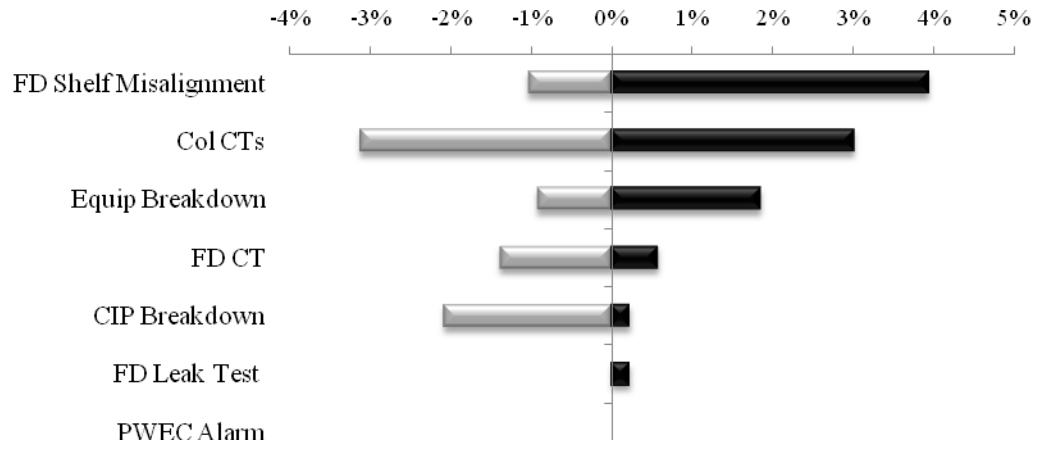


Figure 4.8 Impact of key uncertainties on completion cycle time for 40 batches

4.5.2 Scenarios Analysis

4.5.2.1 Scenario analysis setup

The setup of the scenario analysis comprised of two parts. Firstly, the scenarios were selected based on the results of deterministic analyses. Secondly, taking those parameters with the greatest impact on the completion cycle time from the deterministic analysis, randomness was added to the model during each scenario run.These parameters are shown in Table 4.6. The chosen scenarios are discussed below.

The scenarios chosen were as follows:

- Additional operator
- Additional freeze dryer
- Additional water tank
- Change to start of buffer makeup

The addition of the operator was chosen as a scenario due to the findings discussed under section 4.5.1.1 which showed that labour was a constraint on the process, causing activities delay as labour became unavailable at various points during the

process. The findings also showed the impact, on the process utilisation, of adding just one additional operator.

With regards to the addition freeze dryer, the unit operation could only process one batch at a time with no storage between the final purification stage and the dryer, meaning that the product would not load onto the final column until the dryer was ready. This of course placed a significant constraint on the scheduling of the process and, as Gantt outputs of the process showed, gaps appeared in the process schedule where batches could not be fed through due to unavailability of the freeze dryer suggesting that itwas a major bottleneck. This could only be solved by significantly reducing the cycle time of the step (rather unlikely) or by increasing the number of freeze dryers.

In order to understand further bottlenecks in the process, the series mode of operation was removed, allowing overlap of batches in the Backend Process. The purpose of this exercise was to determine what would be a major constraint if the process were allowed to run as fast as possible, with only the resources acting as constraints i.e. in parallel mode. As expected, the series mode output Gantt chart showed no overlaps in the start-finish of each batch. Although it was expected that there would be overlaps in the parallelmode Gantt chart, it was in fact identical to that of series mode. This anomalous result in parallel mode however was not due to the freeze dryer as that would not stop the second batch from starting before the first had finished. It would instead show an overlap in start-finish followed by a waiting time represented by a gap between the preparation of the final column and its running (similar to the series case). Since the process was being pushed into series mode there must be another earlier constraint causing it to do so. Review of the model outputs showed that the first column was delayed in preparation because it was waiting for Water, classified as one of the BackEnd buffers. Only one tank was used to makeup the water supply, with six make-ups necessary to process one batch. In series mode the low makeup cycle time meant that it did not act as a constraint, with enough time to meet the water requirements of the one batch before the next came through. However in parallel mode the second batch required water while the tank was still being used elsewhere, hence the delay in preparation of the first column.

Therefore the water (PWEC) make-up tank was a bottleneck and therefore a chosen scenario.

Finally, under Section 4.4.4.1 it was stated that buffer makeup for each batch would only begin when the final column (col 4) had been cleaned and placed in storage. This rule was put in place following discussions with the BioSynT process engineers. In order to understand the impact of this rule however, the controlling gate was removed, thus allowing buffer makeup to be a continuous loop process. The simple analysis showed that the earlier availability of buffers had a positive effect on the process cycle time due to the fact that equipment preparation was impacted to a lesser degree with regards to waiting for buffer availability. Therefore the change in buffer makeup start time was chosen as the final scenario.

Table 4.6 Showing the uncertainties incorporated in the model and distributions used where TBF = Time Before Failure, TTR = Time to Repair

Parameter	**Distribution**
Equipment Breakdown	Triangular distribution TBF(3,7,14) days TTR(1,6,10) hrs
CIP Rig Breakdown	Triangular distribution TBF(0.5,1,2) days
FD Shelf Misalignment	Probability of Success through Triangular distribution TBF(0.1,0.25,0.5)days
FD Run CT	Triangular distribution TBF(-10%,base,+10%) hrs
Columns Run CT	Triangular distribution TBF(-10%,base,+10%) hrs

The stochastic simulations were run for 1000 iterations each in order to achieve reasonable convergences.

4.5.2.2 Scenario analysis results

A frequency distribution was then generated for each scenario. Figure 4.9 shows that generated for the base case. Here the distribution is as would be expected with the peak value corresponding with the base case cycle time of 8.62 months, since this base case was obtained using the most likely values. The distribution also shows the range of possible completion cycle times to be between 8.45 and 9 with the best and worst case combinations of uncertainty values. This means that although the probability is relatively low (as indicated by the area under the peak) the cycle time could be increased by 4.4% which is an undesirable event.

Figure 4.10 shows the frequency curves derived from these distributions along one axis in order to compare their positions relative to each other as well as the base case. Only those which had an impact on the completion cycle time have been included.

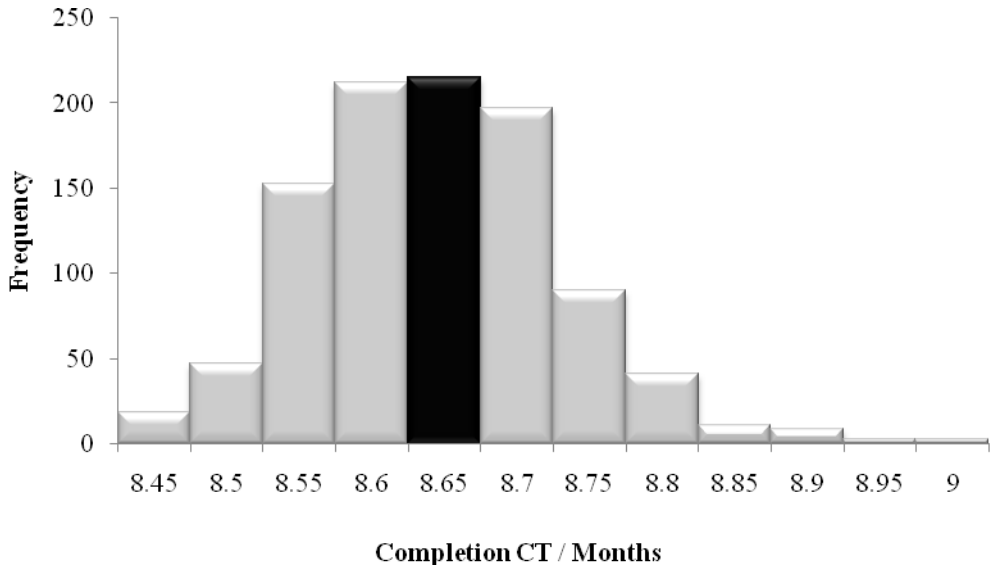


Figure 4.9 Frequency distribution of completion CT for 40 batches of BioSynT in the base case scenario. 1000 iterations were used with a convergence of 0.0001 in the Monte Carlo simulation to account for equipment failures and cycle time uncertainties. The darker shaded column represents the deterministic value of completion CT.

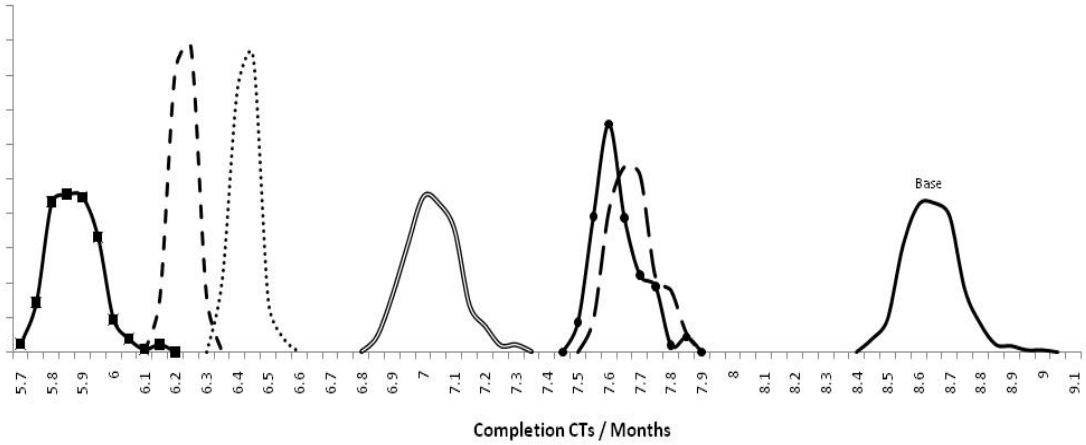


Figure 4.10 Combined frequency distributions of completion cycle times for 40 batches showing the scenarios with greatest impact on completion CT. Each scenario run for 1000 iterations with a convergence of 0.0001. These scenarios are (a) base case (——), (b) additional freeze dryer (— –), (c) additional operator (—|), (d) additional freeze dryer and water makeup tank (—•—), (e) additional freeze dryer and an operator (·····), (f) additional freeze dryer, water makeup tank and an operator (- - -) and (g) additional freeze dryer, an operator and change to start of buffer makeup (—■—).

The stochastic results clearly show the lower cycle times which can be achieved if any of those scenarios shown in Figure 4.10 are implemented. The resolution of the primary and secondary bottlenecks together, that is, the freeze dryer, operator and buffer makeup, result in a reduction of several months for the processing of the 40 batches, bringing the cycle time to below six months. In fact most of the peak range lies below the 6 month point meaning that even with the uncertainties in place the likelihood of achieving a 6 month process time is high.

This chart also supports the ranking of the bottlenecks, that is, the labour availability as the biggest, the freeze dryer cycle time as the second and the buffer makeup trigger as the third. This sequence is shown by the position of the peaks relative to each other; the more significant the bottleneck the further away from the base peak.

To take this analysis one step further it would be interesting to quantify the findings in terms of the investment that would be required if the changes highlighted by the bottleneck analysis were made. Figure 4.11 is a bubble diagram illustrating this.

Taking the standard deviation to be the risk involved, the preferable place to be on the bubble chart would be in the bottom left hand corner, that is, low risk and low completion cycle time. According to the chart the scenario with FD+Water+3Ops would therefore be the better choice. However what little is gained in terms of cycle time reduction relative to the next scenario, FD+3Ops, does not really justify the extra £30k pa investment. Therefore of these two scenarios, the latter would be preferable. But then the scenario with FD+3Ops+Buffer Makeup although has a higher risk, has the lowest completion cycle time. This is highly beneficial if the company want to close the facility down for half the year or dedicate it half the time to another product. If however the minimum investment is desired for reasonable gain in cycle time reduction, then perhaps the best scenario would be to appoint the one additional operator. This would reduce the completion cycle time to just below 7 months, allowing for facility shutdown (or use for another product) for almost 3 months out of the 10 month time horizon.

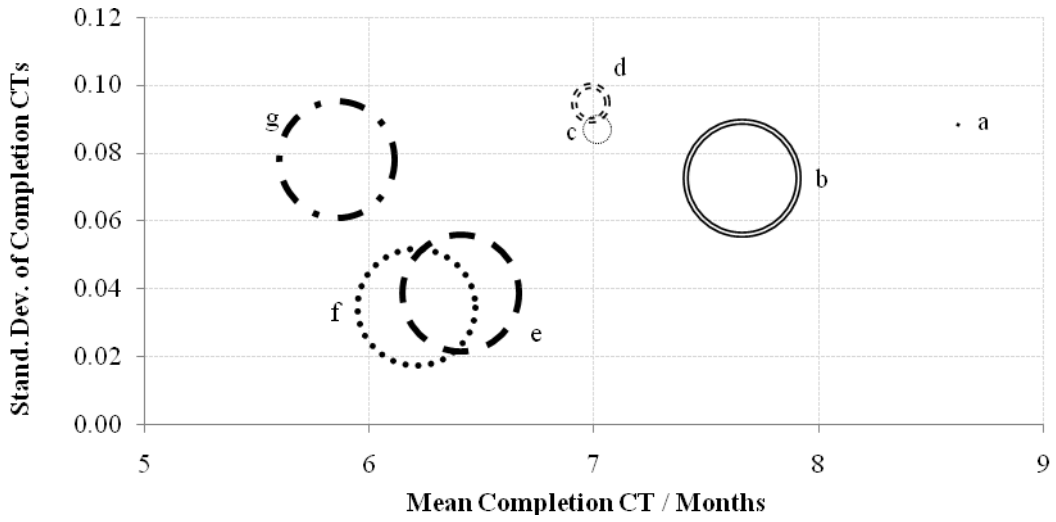


Figure 4.11 Bubble Diagram showing mean completion CT v. standard deviation for different scenarios: (a) base case (•), (b) additional freeze dryer (═), (c) additional operator (····), (d) additional water makeup tank and an operator (= =), (e) additional freeze dryer and an operator (— —), (f) additional freeze dryer, water makeup tank and an operator (····) and (g) additional freeze dryer, an operator and change in trigger of buffer makeup (— · —). The bubble size is proportional to the investment required for each scenario. Input costs: (1) Freeze Dryer £500,000 (capital), (2) Water tank £15,000(capital), (3) Additional Operator £45,000pa (£30k plus 50% overhead costs (Coulson & Richardson 2005)).

4.5.3 Evaluation of use of the Standard Framework in Construction of the BiosynTModel

The BioSynT model was built to answer the specific capacity management question of how fast could the process be made to go and also to test the Standards Version 1 framework its ability to guide the construction of a model that meets the set requirement specifications. In order to understand and evaluate the value of using the framework to construct the BioSynT model a qualitative analysis must take place. In an ideal world, a quantitative analysis would also be performed however it is exceedingly difficult to quantify the comparison between the process of two construction methods. What can be offered is the feedback from the end users and the validation outcomes of data/trend verification, using the requirement specifications as a guide.

Intuitive to user

According to the guidelines outlined by Valentin and Verbraceck (2002), a model can be made intuitive to the end user if it is constructed in such a way that the 'Interactions between model parts...should represent interactions in the real system', 'Use concepts that represent functionalities as found in reality and that can be used for visualisation purposes' and 'Visualise a system in such a way that complexity is reduced but the essential processes are still shown'. These statements simply mean that in order for the end user to fully understand the model within a short space of time, the system elements should be easily identifiable within the model and that any model elements which are non system specific should be hidden and ideally minimised so that the level of complexity is reduced. Following these guidelines, the BioSynT model was constructed using functional decomposition and therefore the main block constructs visible to the user are the main system operations. Furthermore, the hierarchical nature of the Extend platform allows for model elements to be hidden away within blocks, helping to reduce visible complexity. This feature was heavily utilised in the BioSynT model.

Relevance and Ease of Data Input/Output

The BioSynT model uses the SDI link offered by the Extend platform whereby the database source sits within Excel. All of the parameters needed to run the model have been pre set into the database meaning that the user need not access the input Excel

file. The reason for limiting the access requirement is due to the fact that the Excel database would have to be manually exported and imported in order for any changes to be applied in the model database, no matter how small those changes. Furthermore, the SDI link is designed with a rigid structure meaning that in order for Extend to be able to read the tables correctly any structural changes such as an extra row or field in any table would have to be manually noted within the Excel file before exporting. These features are deemed as shortfalls because they increase the necessity of user input where knowledge of the model and database is required. As such the major variable parameters have been placed within the model using the intuitive notebook feature so that the user can easily access them. However, the nature of the database and the limitations of the notebook function mean that the degree of variability is reduced, with parameters such as cycle time only being changeable within the relatively complex database structure.

Maximised reusability and sustainability

Table 4.7 shows those parameters easily accessible to the user. The model inputs clearly shows the limited number of parameters made truly variable, with all other parameters accessible only through the database. As stated earlier, these latter ones can be changed by the user but only if they have knowledge of the SDI function. Although the model has been set up in such a way that the user can easily change the number of equipment and labour, further scenarios analysis would need access to this database and possible structural changes to the model. For example, a change in the process sequence would need alterations to both model and database. Although rather difficult to quantitatively score the BioSynT model in terms of reusability and sustainability, it is possible to qualitatively state that these requirements are limited.

Table 4.7 Summary of inputs and outputs to BioSynT model

Model Inputs (Notebook)		**Model Outputs (Excel)**	
Resources	- Number of labour available	Throughput	- Batches
	- Number of equipment	Durations	- Activities
		Utilisation	- Equipment, Labour
		Other	- Ticket generation
		Graphical representations	- Excel Gantt charts

Short Run-time

The run time of a model is important particularly when the number of runs required increases. For example, if a model were to be run only once, then even a ten minute run would not be too significant. However if 1000 runs were required (during a Monte Carlo analysis for example), even a two minute run time would result in two thousand minutes or 33 hours of simulation time. If only four scenarios were then run (as in the case of the BioSynT case) then this would mean 5.5 days of simulation run time. This would not only be draining on CPU capacity but would make the use of the model as a decisional tool somewhat inefficient. Therefore a short a run time as possible, preferably less than a minute, would be ideal. Without an equivalent BioSynT model to perform a direct comparison with, it is rather difficult to quantify any improvement in run-time. However, during the review of existing models it was found that most took several minutes to run only once. The BioSynT model takes approximately a minute and therefore it can be deemed to meet this particular requirement to a certain degree. It is important to note here that discrete event models, due to their nature, take longer to run than purely algorithmic based models and therefore, although run-times of only seconds would be preferable, the complexity of discrete event makes it more unlikely if larger systems are being modelled.

Minimised development time

The major shortfall in the construction of the BioSynT model was the development time; although in theory it should have taken a matter of months in total, the design to final documentation stage took over a year to complete. One major factor was the model rebuild which took place months into project commencement due to the realisation that the model did not quite meet the scope. Section 3.5.2.1 explored this in greater detail.

4.6 Conclusion

The BioSynT case was used to demonstrate the ability of a model, built using the standard, of being used as a decisional tool. A deterministic study was first carried out in order to determine those parameters whose variability would significantly

impact the output metric, completion time for the processing of forty batches, thus identifying the primary bottlenecks within the process. It showed that the biggest impact was seen with the frequency of freeze dryer misalignment, lengthening the process completion time when increased.

A stochastic study was then carried out using the results of the deterministic analysis, looking at the impact of different scenarios combined with process uncertainties to determine how these would affect the running of the process and to highlight any shifts in bottlenecks. A cost analysis of the scenarios was also carried out to add a further dimension to the study. The outcome of this was that in order to reasonably reduce facility operation time whilst injecting minimum investment the best scenario would be to increase the number of operators by one, giving a reduction in overall completion time of almost 3 months.

Finally the BioSynT case study was used to implement the Standard Framework 1, testing its ability to guide the construction of a manufacturing capacity management model capable of meeting the requirement specifications stated under the standard. A qualitative analysis showed that these were not sufficiently met, with rigidity in the method of model construction and the approach used, leading to a less than adequate reusability and sustainability, a relatively long run time and a far greater development time than desired (with time and cost efficiency as measures).

CHAPTER 5

Application of Standard Framework to a Biotechnology Capacity Management Case II

5.1 Introduction

As illustrated in Chapters 3 and 4, version 1 of the Standard Framework for building capacity management models required improvements to optimise development time, model reusability and extensibility. This chapter illustrates a case study, the large scale production of a monoclonal antibody, used to test the ease of use and limitations of version 2 of the Standard Framework when addressing industrially relevant questions. The case study focuses on how a biotechnology facility built for the commercial production of monoclonal antibodies will cope with future cell culture titres and searches several strategies to identify the optimal set of process changes to make in order to overcome expected capacity bottlenecks.

The reason for selecting this particular case study (known as mAb) was three fold; firstly, having based the standards amendments on a biotechnology process (Chapter 4) it made sense to illustrate the new version on another biotechnology case in order to truly test its applicability in biopharmaceutical capacity management. Secondly, the case offers new dimensions to those examined in the Chapter 4; particular modelling challenges include capturing split batches occurring in parallel across multiple unit operations, enabling brute force combinatorial optimisation and capturing the consequences of failure rates and including costs. These modelling

challenges are expanded on in greater detail in Section 5.3. Thirdly, to provide an immediate data source which would speed up the data gathering process for any new model, two models, one at the facility level and one at the process level, existed. However having been built without the aid of a standard framework these models were deemed to fall short of the requirement specifications. Furthermore the original scopes had developed and therefore a new model would be needed to cover the new scope.

Section 5.2 provides a brief background to the mAb case study, outlining the reasons for selecting this particular process in demonstrating the use of the framework tool. Section 5.3 will discuss the method behind the case study, looking at the stages of the model design and construct. In Section 5.4, the results and discussions section, a deterministic analysis of model parameters is presented, using process throughput as the objective function. In this section, Monte Carlo simulations are also used to mimic the randomness of one particular parameter highlighted as high impacting. This section also discusses the scenario analyses used to determine the best strategy to deal with process variability and uncertainty while minimising the downstream cost of column repack and maximising process throughput.

5.2 Case Study Background

A case study examining the implementation of version 2 of the standard framework to a biotechnology process will now be presented. The case is based on a new facility built for the large scale manufacture of mAbs. The main drivers for a capacity management model are:

- The process has been designed to cope with a certain titre range of 1.5g/L ±10%, however it is expected that titres up to 10g/L could be seen. This means that there are concerns over downstream bottlenecks since the main constraint to process throughput is the binding capacities of the downstream chromatography units, in particular the initial Capture column step. Possible solutions include greater numbers of cycles (splits) in these units or upgrades in equipment size or resin type. However there are associated tradeoffs between upfront investments costs, running costs and efficiencies which must be investigated.

- Due to the downstream capacity limitations, it is necessary for the harvest stream to be split into cycles once it has existed the recovery area. This is so that both the volumetric and binding capacities of the chromatography units can cope with the product stream. A question posed here is what number of splits will reduce the downstream bottleneck while remaining feasible and cost effective; the tradeoffs being greater step efficiency but greater costs associated with resin replacement (the greater number of cycles means more repacks required).
- Equipment failure, both mechanical and contamination, is a concern both in terms of subsequent shutdown periods and the discarding of valuable batches or batch lots. An understanding of the impact of variable failure and discard rates on process throughput will highlight a threshold beyond which process efficiency drops below an acceptable value.
- The facility is due to begin producing a single mAb product however it has been built with the intention of introducing more mAb products as they emerge from the clinical trials pipeline which will use the same platform process. The impact of this is that although the case study will focus on a single product scenario, the model constructed should have multiproduct capabilities in order for reusability and sustainability,

Initially, the existing models were reviewed and, comparing the new scope with those of the old models, it was deemed that only one model would be required, providing a sufficient level of detail necessary to fully answer the capacity management questions being asked. With a finalised scope in place, the model was constructed using the standards framework as a guide to ensure that it met the model requirement specifications, proposed in Chapter 2. The output parameters were then validated with the mAb team at Eli Lilly to ensure that the model had realistically captured the elements of the system such as scheduling constraints and operating parameters. A deterministic sensitivity analysis was then performed to identify the key factors influencing the process throughput. Sensitivity scenarios were then carried out to examine the effects of higher titres on the process throughput and to determine the strategies which could be adopted to deal with any bottlenecks identified. These scenarios also serve as a demonstration of the capabilities of the model in answering biopharmaceutical manufacturing capacity management questions and therefore, also demonstrate the effectiveness of the standard

framework in constructing models with such capabilities. The following will describe the stages of model construct and the setup and results of all the analyses.

5.3 Method

5.3.1 Model Development

The following sections describe the construction of the mAb model using the Standard Framework Version 2, with the main stages: Problem Structuring I, Problem Structuring II, Design and Construct.

5.3.1.1 Problem Structuring I - Scope

The first part of the problem structuring phase is where the project scope is defined and the model is characterised. In the mAb case study, two levels of questions and requirements were identified under the project scope, at the facility level and at the process level. These are listed in Table 5.1. Table 5.2 gives the required model outputs.

The scope of the case study combines and expands on the scope of the two existing models. Although these models were originally built based on different scopes, they converged to capture similar elements. However they failed to meet key requirements identified in the Standard Framework, namely, ‘intuitive to the user’ and ‘reusability and sustainability’ thus making them difficult to navigate and extend. Hence the need for a new model, built with the aid of the standard framework which could meet these requirements while addressing the new extended model scope.

Table 5.1 Scope given for mAb model

Category	**Question**
Scheduling	- Single product campaigning based on scheduled preventative maintenance shutdowns - Trigger points for buffer and media makeup - Downstream split/combine ratios - Implications of cell culture contamination on the schedule - Understanding of the material holding times - Implications of staggered formation of 5000L fermenters
Resources	- Equipment sizing - Utilisation of various resources (equipment, utilities, buffers) - Required WFI generation rate - CIP skids utilisation and utilities requirement of CIP and SIP - Variability in resource/utility requirements - Volume of media required for each growth step - Labour usage and number needed
Debottlenecking	- Location of bottlenecks
Future capabilities	- Multi-product campaigning based on constraints and influencing factors such as different turnaround procedures - Implications of different trigger points for new campaigns in multi-product campaigning

Table 5.2 Required outputs of mAb model

Category	**Outputs**
Throughput indicators	- Batch throughput - Base grams throughput - Completion cycle times
Resource Utilisation	- Minimum number of labour required to run process - Utilities utilisation profiles - Sizes and number of equipment

5.3.1.2 Problem Structuring I – Model Characterisation

There are two parts to the characterisation of the model. The first considers the type of system and the second defines the modelling platform. Firstly, the system being looked at is a production process for monoclonal antibodies therefore it is quite clearly a manufacturing system. Secondly, in order to determine the modelling platform it is necessary to summarise the scope according to the criteria for platform selection. Table 5.3 shows this summary.

Table 5.3 Summary of new mAb model scope

Feature	**mAb**
Single or multiple activities	Multiple
Continuous or discrete?	Discrete
Static or Dynamic?	Dynamic
Stochastic?	Yes
Mass balancing?	Yes basic
System elements	Entities, Activities, Resources
Metrics	Overall CT, Throughput (batch, base grams)
Constraints?	Yes

Table 5.4 compares the four major types of the simulation tool, against the requirements of the mAb case study as defined under the project scope. The comparison has been carried out using a scoring system where relevant, representing the extent to which each platform is able to model each requirement.

Table 5.4 Showing the different types of modelling platforms and their modelling capabilities in terms of model scope requirements.
Scores given out of 5, 0=not at all, 5=fully meets requirement

	Spreadsheets (Excel)	**Mass Balancing** (Batch Plus)	**Discrete Event** (Extend)	**Equation Oriented** (MathCAD)
Single or Multiple Activities	Single (too complex at multiple)	Multiple	Multiple	Single
Continuous or Discrete	-	Discrete capabilities (see below)	Discrete with continuous capabilities	-
Static or Dynamic?	Static	Static (Aspen Plus Dynamics can extend static models into dynamic)	Dynamic	Static
Stochastic capability?	3	4	5	3
Mass balancing?	4 - statically	5	3- basic	4 - statically
System Elements	3 - Not dynamically	4	5	2 - Not dynamically
Metrics	0 - Outputs required are based on dynamic calculations	3 - Better for platform defined metrics e.g. biochemistry	5	0 - Outputs required are based on dynamic calcs
Constraints	1	3	5	1

Since the mAb case is a discrete event one, this immediately eliminates spreadsheets and engineering/mathematical platforms. This leaves mass balancing and discrete event tools. The scope requires basic mass balancing which would make both Batch Plus and Extend viable candidates. However the mass balancing is only basic and more emphasis is placed on user defined metrics and the capturing of all system elements and constraints. Therefore it is believed that Extend would be more suitable as it meets all of the requirements.

In conclusion the model can be characterised as a *Manufacturing Extend* model.

5.3.1.3 Problem Structuring II

The second part of problem structuring builds a non-coded description of the process. This means that text, diagrams and spreadsheets are used to create a comprehensive picture of the process using only process (and not model) terminology. Problem structuring is largely platform independent and uses the predefined list of questions from the standards framework as a basis. This stage of the construction process was particularly emphasised in the mAb case study in order to understand and validate the process and data before actually building the model. This greatly helped to avoid a rebuild. Figure 5.1 shows a flowsheet diagram of the mAb process followed by Table 5.4 which shows the general system parameters identified as important to the running of the model. It is important to emphasise the importance of this table as it forms the basis of all calculations carried out within the input database and provides a global reference for scenario selection within the model.

It must be noted that this table does not show all of the data required for the running of the model, only the important parameters used for global referencing. Each process step has its own data table detailing its specific resource and CIP requirements, sub tasks and cycle tines.

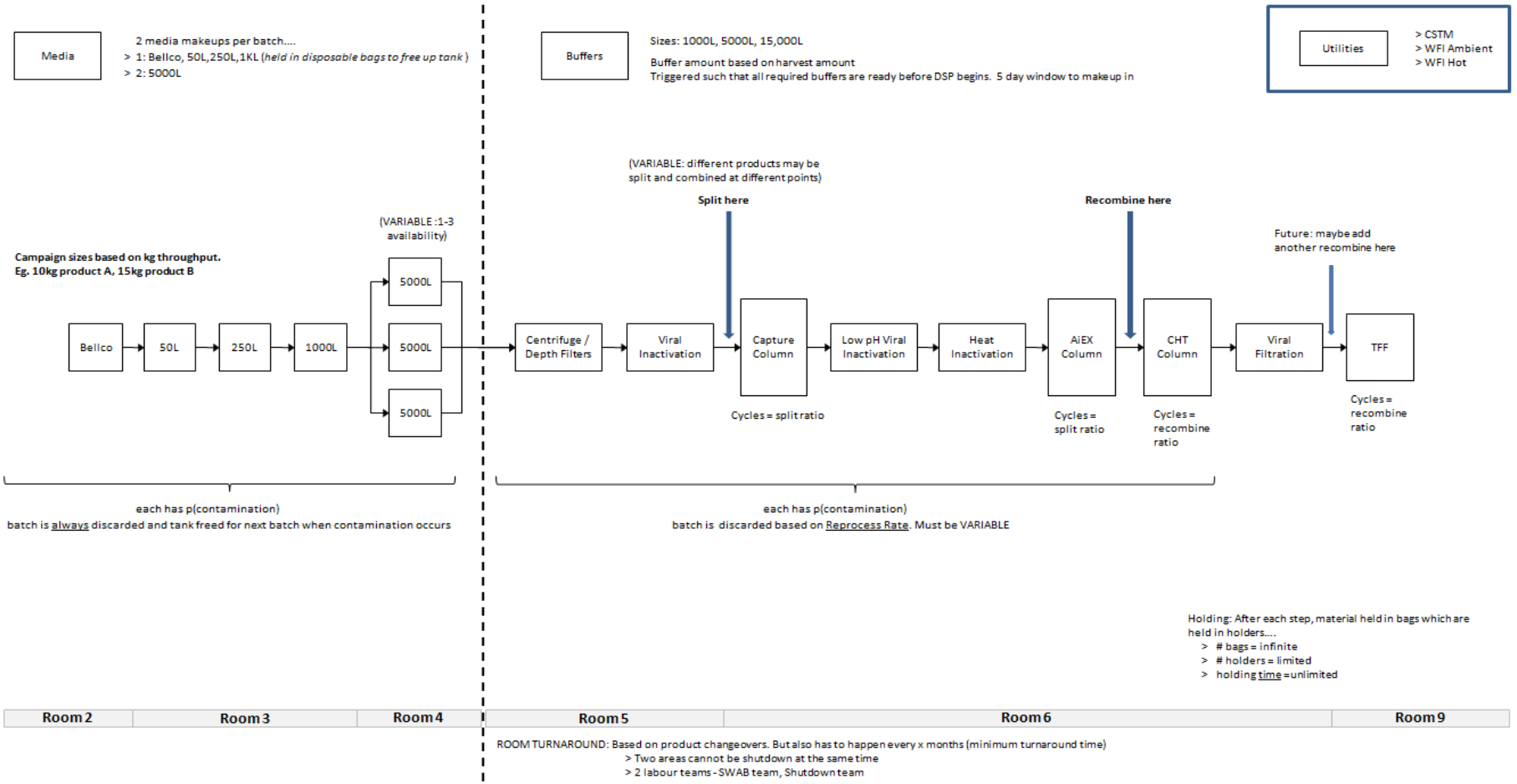


Figure 5.1 Flowsheet diagram of mAb process comprising fermentation train, recovery and chromatography based purification operations. Key media, buffer and utilities resources are also shown. An indication of suites for each operation is illustrated.

Table 5.4 General parameters used as inputs to mAb model

Parameter	Description	
Number of operators available	Number available per shift defined under Scheduling	2
TPrep for Media	Window in which to prepare media before it is needed (hrs)	96
Cycles Col-CAPTURE	Number of cycles after which column needs repacking	80
Cycles Col-AEX	Number of cycles after which column needs repacking	80
Cycles Col-CHT	Number of cycles after which column needs repacking	80
Recovery Yield	Calculated during initiation	90%
Purification Yield	Calculated during initiation	75%
SD/Turnaround window	Time between current Turnaround requirement and next scheduled Shutdown where synchronisation is allowed (months)	0.5
Split Ratio Calculation Method	Based on titre, not limited = 1, Based on titre, limited = 2, Based on user input = 3	2
Rolling shutdown	Rolling shutdown = 1, Synchronised shutdown = 2	2
Height Calculation	Fixed = 1, Based on Titre = 2	1
Titre entry	Fixed = 1, Random based on triangular distribution = 2	1
Cycle Limit	Cycle limit or split ratio limit for CAPTURE column	8
Split Occurrence	Equipment ID before which split occurs	13
Combine Occurrence	Equipment ID before which batch combine occurs	17
Shift Duration per 24 hrs	Hours during the day that are covered by shifts	18
Number of Bellco		1
Number of 5K Bioreactors	Enter 1-4	3

5.3.1.4 Design

The design of the model was very much in adherence to the guidelines laid out in Standards Version 1. That is, with the IDEF0 defined left-to-right flow structure, the functional decomposition of the main block constructs whereby blocks represent activities (rather than equipment) and the use of templates where possible. This has resulted in main block constructs which all look very similar, apart from small variations due to the nature of the activity. For example, chromatography columns have cycles whereas fermenters do not. These cycles are represented by the splitting of the process stream into the equivalent number of cycles in order to process the batch. This feature is explained in further detail in the next section. The majority of the template blocks have been constructed so that they may be extracted and used for any process with similar attributes such as cycle times and equipment. These templates and their role in the standard guidelines is discussed in greater detail in Chapter 3.

Table 5.5 gives a more comprehensive mapping of the system and model elements. The detail behind the model design, for example the block constructs, the templates used and the features captured in discussed in the next section.

5.3.1.5 Construct

The construct phase is very much linked to the design phase and in fact the two occur simultaneously, with the design of the model guiding its construct. During this phase of the mAb case study, the templates were placed in their appropriate positions and tailored to various degrees in order to truly represent the process elements as in the case of the cycles within the chromatography blocks. During the construct phase it was a key aim to reduce the visibility of Extend blocks. In other words, the first four hierarchical levels down would only show hierarchical blocks, rather than actual Extend blocks, representing either logic or process elements, thus helping user intuitiveness, should the user ever need to access the actual model.

Table 5.5 List of the system and model elements for the new mAb model

	Element	Mapping
System Elements	Activities	Process steps USP & DSP Buffer/Media makeup Pre, Run, Post
	Entities	Batches / Multi-product
	Resources	Labour Equipment Ancillary: CIP, Utilities Rooms
Layout	Physical Layout	Functional decomposition
	Logical Layout	Process flow
	Parallel Activities	Combined functions
	Main Block Construct	Process steps
	Item Transfer	Blocks and arrows
	Item Flow Control	Gating
Initiation	Items Generated / Primary Item	As fast as process is capable. Not scheduled
Model Logic	Metrics	Mass balancing All Resource Utilisation Hold times Failures –stochastic Load Volumes Equipment sizing Split ratios
	Look ahead	Clean of previous tank Prep of next tank Buffer / media makeup
	Shifts	Different shift patterns
Data	Data transfer	To Excel via Extend database
	Database	Template based
	Tracking	Global arrays and database

Layout

Functional Decomposition was used and therefore the hierarchical blocks represented the activities in the real system. The top level of the model looked like that in Figure 5.2 with all 'storage' blocks (blocks containing externalised elements, global arrays

and resource pools) positioned at the very top of the workspace. Figure 5.3 shows the next hierarchical level and the main activity blocks.

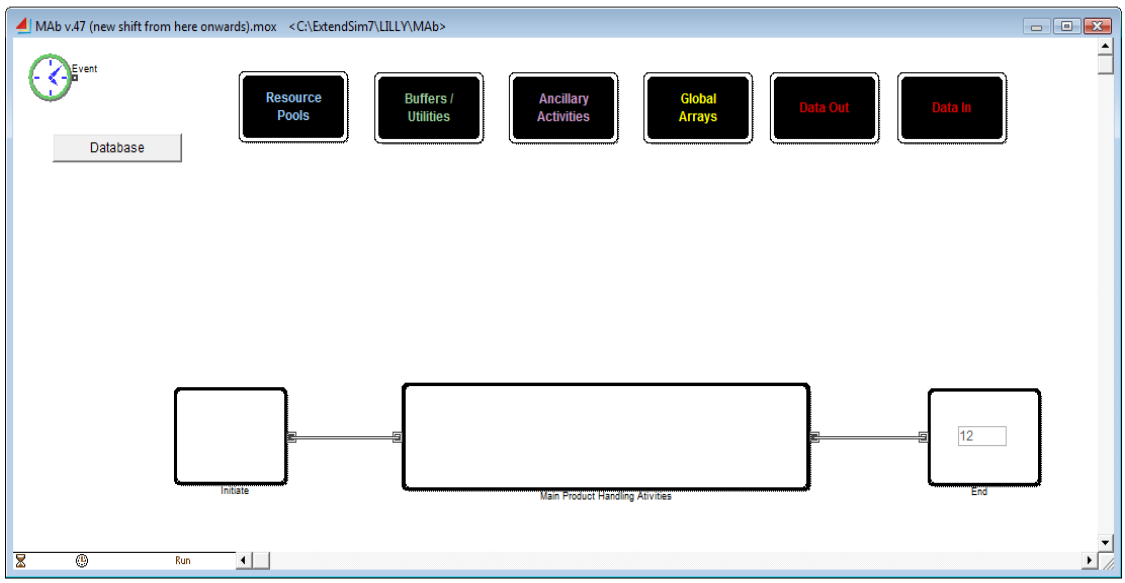


Figure 5.2 Top most hierarchical level of mAb model

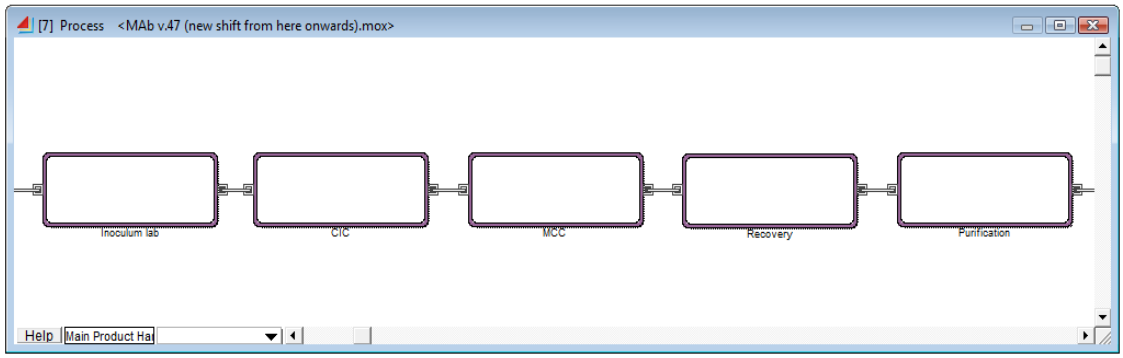


Figure 5.3 Second hierarchical level showing main activity blocks

Main Block Constructs

Inoc, CIC and MCC

With the exception of Inoculum growth, all of the fermenter blocks were identical in structure with five pre-run sub- activities before the actual run. The selection of the 5000L vessel occurred before the professing of the 1000L fermenter within the 1000L main activity block and was based on the availability of the each of the 5K fermenters. Based on selection, that fermenter's preparation was triggered using the standard Look Ahead block, placed here rather than in the Initiation block due to the fact that it was not known which 5k fermenter would be used until this point.

Recovery

As the preparation of the diafiltration system was carried out along with the preparation of the centrifuge the preparatory sub-activities were only present within the Centrifugation activity block. Furthermore, unlike the other main activities, the diafilter went through CIP before its last sub-activity which was the removal of the filters.

Purification

The three chromatography columns, Capture, AeX and CHT, were very similar in structure and had the same cycle feature. Each column also had a repack check block which checked if repack was required based on whether it had exceeded the maximum number of cycles allowed or whether there would be a product changeover. If repack was required the equipment was sent to the Repack block which is described below.

Repack

Repack of a column was an activity which occurred externally under the Purification hierarchical block. When the repack check determined that a column required repacking it was thrown to this block and once the column had been repacked, it was sent back to its own hierarchical block where the repack time was reset.

Buffer/Utility Blocks

Buffer Makeup

There were 3 buffer makeup vessels of different size, 1000L, 5000L and 15000L. The makeup activities in each were identical, they were simply chosen based on volume of buffer to be made up and the availability of the vessel. The logic used to select also ensured that of the possible vessels available and able to make up the volume, the smaller of the two was always selected unless not available.

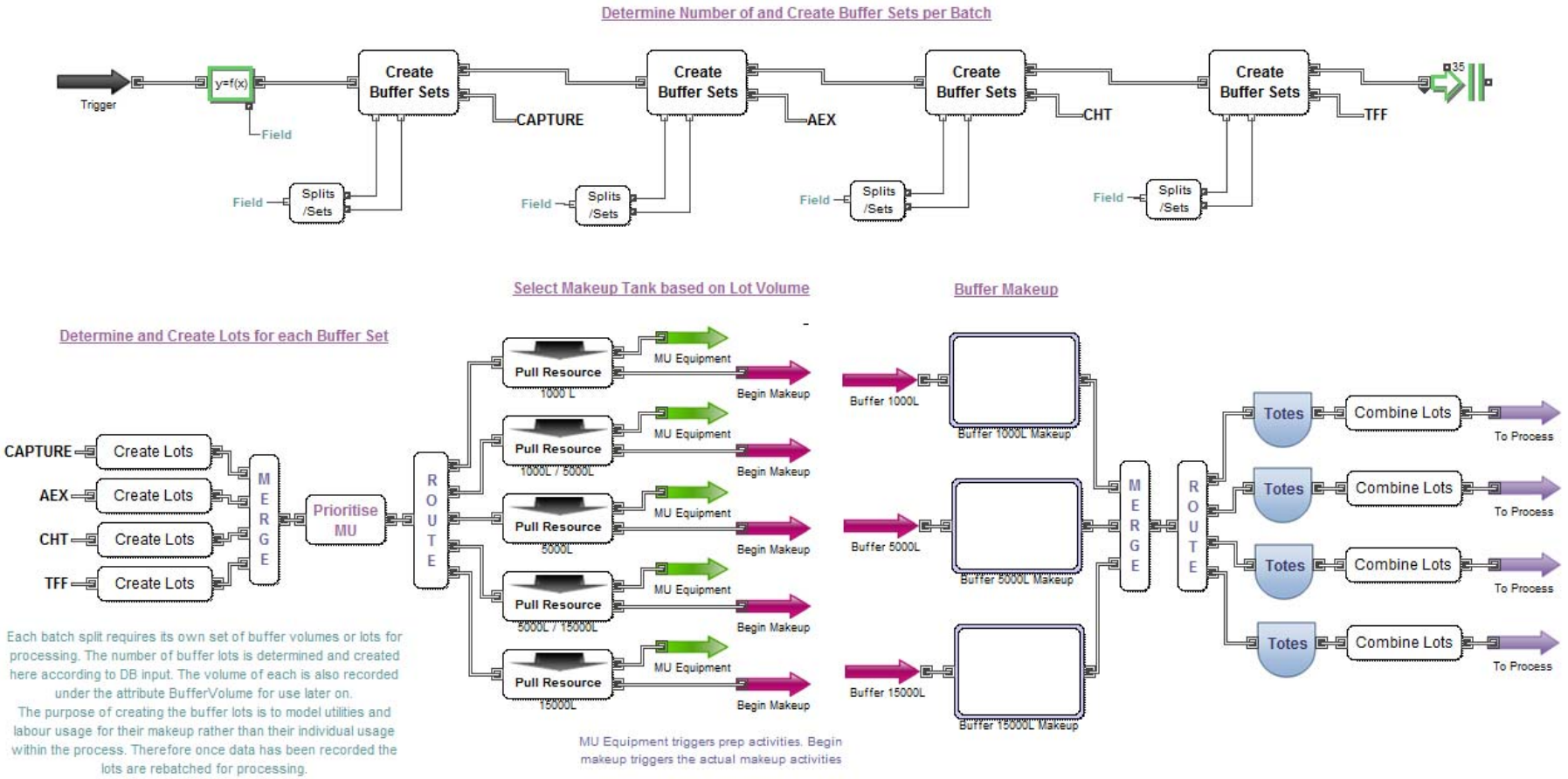


Figure 5.4 Buffer makeup hierarchical block showing the receiving of the trigger item, determination of the buffer sets, creation of buffer lots, allocation of makeup tank based on volume and buffer makeup before storage in totes prior to routing to process.

Upon opening the Buffer hierarchical block (Figure 5.3) there were several sections immediately visible. The first, at the top of the page, created buffer sets where the number of sets equalled the split ratio or the number of cycles through each column. The next section was where buffer lots were created based on the fact that each cycle through a column required a buffer set made up of buffer lots so that

$$\text{buffer set} = \sum [\text{buffer lots}] \tag{5.1}$$

Each buffer lot had a different volume depending on the product and the column (the volume which determined the choice of makeup vessel). The different volumes could be found in the database and were variable.
So, for example, for the product being modelled in this particular case, the capture column required 8 buffer lots for every cycle. If the split ratio was 6 then there were six cycles through the column. Multiplying the numbers gave the total number of buffer lots for that batch i.e. 54. The 'Route' block then sent the buffer item to a vessel request block based on the volume and the vessels capable of handling that volume. Once the vessel had been pulled from the resource pool, two items were thrown to the buffer activity block; the first triggering preparation of the vessel, the second representing material processed. Once buffer was made up, it was held in disposable totes and then routed to the process main stream.

Media Makeup

There was only one makeup vessel for media, 4000L which was used to make up both 1000L and 4000L media volumes, made up in that order. When 1000L was made up it was stored in disposable totes to free up the tank for the next volume. When 4000L was made up it was held in the tank until required by a fermenter.

Utilities

Three utilities were modelled: (1) CSTM for SIP, (2) WFI Ambient for buffer makeup and (3) WFI Hot for CIP. The source supply had infinite capacity however the valve controlled the filling up of the holding tank, only allowing it to be filled when its contents dropped to 25%. The capacity of this tank was set in the database and was variable. The diverge block then routed the flow to whichever area requests

that utility and therefore flow was determined by demand. This allowed for the required generation rate to be determined.

Ancillary Activities Blocks

There were two CIP skids:

- Skid 1 used for Upstream and Recovery
- Skid 2 used for Purification and Makeup

Based on the allocation of the skids to the various process equipment and the CIP requirements, the CIP block had four main streams. The CIP requirements were as follows:

- USP: after each harvest there was a sequence: Rinse Only → SIP → CIP
- Recovery: CIP only
- Purification: CIP only
- Makeup: the makeup tanks went through Rinse Only by default unless the CIP expiry was exceeded or the maximum number of rinse cycles had been exceeded (variable)

All CIP/Rinse blocks were identical. Each block could be set to Rinse Only, CIP only or either of the two by entering a 1,2 or 0 respectively into the top left hand corner dialog box.

Other Blocks

Split/Combine

The splitting or combining of batches and lots was a highly variable process as the number of splits and the location of both split and combine were user defined (the number of splits could also be set to calculate based on titre). This meant that all activities where they could occur must contain the appropriate blocks i.e. all downstream activities. The modelling method used comprised of a check which looked up the database split/combine table. This table gave both the ratio and the location given as activity ID. If the lookup logic recognised that for example a split was to occur, the batch was split into the correct number of cycles for processing. A similar procedure for combine occurred where the correct number of lots wererebatched. A particular challenge faced here was the combining of lots within a

batch where one or more lots had been discarded. In order to deal with this, a global array was set up to track the number of batches and lots being processed at any point in the process at any time. Thus if a discard occurred in an activity, the number of lots being processed there and in all downstream steps would be decremented by one. The activity where combine would then occur would be able to reference this array and know that a fewer number of lots than expected (according to the combine ratio) would have to be rebatched.

Area Shutdown/Turnaround

Turnaround occurs between product changeovers and this is therefore more relevant to multi-product scenarios. However as it is an important feature of the model's capabilities, it is discussed here. The last main activity of each process area or room contained an Area Shutdown/Turnaround block. Shutdown is simply scheduled shutdown which was set to every 6 months although this was variable. These two activities were entirely separate but could occur together if scheduled that way; within the initial Route block several blocks could be found which decided this. Firstly the equation block decided if turnaround was required based on whether there was going to be a product changeover. The second equation block then made the following decision.

- If turnaround was required
  If the next scheduled shutdown was due within the next 2 weeks (variable), then shutdown was moved forward to coincide with turnaround
- If turnaround was not required
  Shutdown would occur now only if the scheduled period of 6 months had been reached

Once the decision had been made, whether shutdown, turnaround or both, the item was routed to the Shutdown/Turnaround block where the associated activities would take place. Note that shutdown of an area could be synchronised with other areas in which case it would not occur until all rooms were ready for shutdown. This was achieved by placing gates ahead of the shutdown area, all referencing a global array where the activity status of the rooms was stored. Once all rooms to be shutdown had been marked with 'ready', the procedure would go ahead by opening the gate for the item to enter.

Once the process area had gone through the necessary shutdown/turnaround activities the room and equipment were freed up for the next batch.

Equipment Failure

Each equipment within the main process stream had a run sub-activity which models variable equipment failure. The probability of failure and the subsequence probability of forward processing (whether the batch/lot was discarded or not) were set in the database table and referenced here. If a discard did occur, then the global array tracking the batch/lot throughput would decrement the appropriate array address (based on activity) by one. Figure 5.5 shows a representation of the modelling methods used.

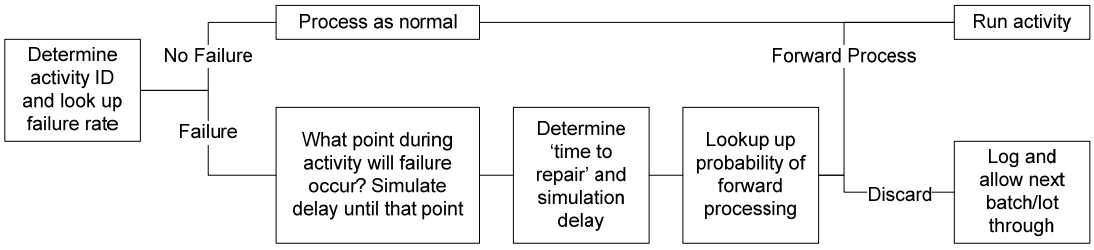


Figure 5.5 Representation of modelling sequence in failure block

Mass Balance

Every main activity block in the process chain contained a Mass Balance block. These blocks calculated the kg throughput of each batch based on the step yield and also the stability of the product, affected by how long the product had been made to wait before moving on to the next step. The Mass Balance block within the Capture column differed from the rest in that it offered the choice of calculating throughput according to stability or number of cycles based on a linear equation, a feature implemented as a 'nice to have' following discussions with the client. Furthermore any lot discards were accounted for here by calculating the mass equivalent of the lost lot or lots within the batch.

Labour Request Process

This block could be found within the Resource Pools storage block and externally processed every labour request made within the model. As a labour request item entered, it waited for the required number of labour. Once it picked up the labour it

then went to the activity block which simulated the labour hours associated with the Sub-activity which made the request.

A challenge posed here was that for the fermentation activities labour was not required for the duration of the fermentation period. Instead the required labour had to be pulled every 24 hours and held for the correct labour hours. This was modelled by using the unbatch block to create the number of labour items which was equal to the number of days. The pulse block then opened the gate every 24 hours to allow one through to process a labour request.

Note that for the 5000L this was taken a step further as different numbers of operators were required each day.

Model Parameters

All model parameters such as cycle times, flow rates, titre, campaign schedule, shift duration, and column dimensions were entered into the external reference file called MAb Input Data.xlsm. This was an Excel file which upon initialisation of the model, was accessed by Extend and used to populate the relevant database tables, in the internal database. The model was set to run for a time horizon of 365 days based on 100% plant efficiency. Any shutdowns were modelled as part of this time.

Primary Items Generated

The main items generated in the model were batches. The required number was generated all at once according to the user defined product demand and their path through the model was scheduled by the gating system and the Look-Ahead logic. Firstly, the Create block was linked to the Production Schedule database table which defined the campaigns generated and the product identity for each campaign. Note that only one item was generated per campaign at this point. The number of batches required per campaign was calculated in Excel according to the demand (kg) and the variable titre using the following equation:

$$\text{Num of Batches} = \frac{\text{Demand (kg)}}{(\text{Titre} \times \text{Fermenter Volume} \times \text{Purification Yield})/1000} \quad (5.2)$$

The result of this equation was then used to generate the required number of batches for that campaign using the Unbatch block.

The BatchNum attribute was then set along with the flagging attributes for the last and first batches of each campaign (although more relevant when modelling multiple campaigns).

Batching and Scheduling

Although a Production Schedule table was used, this only defined the campaign sizes and their start dates and not the exact sequencing of batches. Therefore the model was not based on a production schedule; rather batches were generated to run as fast as possible through the process and their flow self regulated by the rules and constraints modelled mainly via the gating system. The work schedule was based on a 7 day week with 24 hour operation starting Monday 9am.

Metrics

- Cycle Times: Although cycle times of individual activities were set in the database, the actual times varied due to various delays. A simple Timer block was used to record the time at various points in the model. These times were stored under the appropriate attributes e.g. 'BatchStartTime' and later sent to Excel where a Gantt chart was automatically generated.

- Throughput: This was the number of batches within each campaign which were processed and the equivalent base kilograms. This latter value took into account any kg losses due to batch/lot discard.

- Base Grams: This was the kg throughput for each batch. Calculations were carried out at the end of each step and based on the step yield and product stability in storage, the output was determined and stored under the Base_Grams attribute. In the END block, the total kg throughput for the campaign was determined and sent to Excel where it could be compared to the initially calculated required throughput in order to determine the % processed metric, which will be discussed further in Section 5.4.2.4.

Lookahead

Lookahead was a particularly challenging feature as it meant creating a modelling method to decide when to trigger activities based on events which hadn't happened yet and whose occurrence was based on the self-scheduling of the model. The approach adopted was to create Look Ahead blocks which were placed in the initiation section of the model. which were largely identical and were differentiated between by entering the ActivityID of the activity which they triggered preparation of. A table was also created and placed in the database which calculated the cumulative run time of all sequential activities and the preparation time of each based on their pre-run sub-activities. Each block automatically accessed these cycle times to determine the time at which an equipment needed to be prepared in time for the next batch. This time was calculated as a delay which was then used in the Activity block to delay the item before it went on to send a Trigger Item to the appropriate activity equipment hold in the main activity block ahead of all the preparatory sub-activities. Essentially, what this did was to prevent the equipment item from going ahead and performing the pre-run activities before it was needed by having to wait for the trigger item to batch with it. The equation used to calculate the delay time for this trigger delay was as follows:

$$\text{TriggerDelay} = \text{CumRunCT} - \text{PrepCT} \tag{5.3}$$

Where

CumRunCT = Cumulative Run Cycle Times for all Preceding Activities

PrepCT = Preparation Cycle Time for that activity

Data Transfer

Data generated by the model was transferred to Excel using the Data Import/Export blocks.

Database

All data was entered into the built in database within Extend by the appropriate data access blocks which populate it using the external MAb Data Input.xlsm file.

5.3.2 Key Base Assumptions in Case Study

The following are the key base assumptions built into the model.

- The facility adopted a platform process approach and hence the process sequence was fixed for all products entering the facility.
- The base case titre was 1.5g/L, chosen as it represents a typical but relatively low mAb production titre
- The number of available 5000L production fermenters was limited to four. This was based on a facility with the capacity to currently house three of these fermenters with the possibility for expansion to accommodate the fourth.
- Any failure modelled was based on the assignment of the failure rate to all main non-disposable process equipment. Although the base case assumed a failure rate of zero due to the random nature of the parameter, any specific studies assumed a base case value of 4% based on discussions with the process team.
- The calculation of the split ratio (the number of cycles through the chromatography units) was based on the upper range of the titre fluctuation i.e. +20%, assumed and accounted for when specifying the operating strategy of DSP operations (personal communication, Guillermo Miroquesada, Eli Lilly, Indianapolis).
- Split would occur prior to the Capture step and combine would occur prior to the CHT step (final chromatography step) due to volume capacities
- The base case split ratio limit was 8 meaning that no greater than 8 cycles would be passed through the chromatography units per batch.
- Cycle 1 from the Capture step would move straight on to the next chromatography step while cycle 2 was being processed in the Capture column.
- Holding times were not limited as storage availability was an issue being reviewed under the scope.

5.4 Results and Discussion

5.4.1 Sensitivity analysis

5.4.1.1 Setting up the deterministic case

Having constructed the model with certain key base assumptions, described under section 5.3.2., it was decided that these would form the base case scenario against which the sensitivity can be measured. It was also decided that the analysis would be performed at three different titres 1.5g/L, 4.5g/L and 10g/L. The reason for this being that 4.5g/L represents a value that current processes are achieving (Aldridge, 2009) and 10g/L serves to test the sensitivity of the system to products in the future, thus giving a more longer term impact of system variability.

Ultimately, a sensitivity analysis is used to test the robustness of a system to variability. As such, it is important to know which parameters to test, in order to gain a true understanding of their impact. For the mAb process one of the key variable of interest was the split ratio, that is, the number of cycle per batch through the chromatography columns. The reason for this is that the split ratio or the number of cycles is based on the following equation:

$$\text{Split Ratio} = \frac{(\text{Titre} \times \text{Production Fermenter Volume} \times \text{Recovery Yield})}{(\text{DBC of Capture Column} \times \text{Capture Column Volume})} \qquad (5.4)$$

where DBC = dynamic binding capacity. As the titre increases the number of cycles or the split ratio must also increase if the column parameters remain constant. However if that ratio is limited, in this case to a maximum of eight, then the capacity of the column becomes limiting with the percentage of product binding with every column volume decreasing as the titre increases i.e. with increase in titre, a greater percentage of product cannot bind and flows through. This affects the overall process throughput.

According to Equation 5.4, there are further parameters which could also show impact on the % processed value: yield, column dynamic binding capacity and column volume which is determined by height and diameter. Also, the resin lifetime

should also be investigated as it is affected by the number of cycles through a column. If the lifetime is varied then the number of repacks also varies, thus affecting the process cycle time and possibly having visible impact on the % processed due to time availability.

Further to this the failure rate or the contamination rate must not be ignored. The base case scenario assumes that there is no failure within the process however this is actually inaccurate. A certain percentage of failure is present in any new process and therefore must be captured. It can be assumed that the failure rate of a new facility will be in the order of around 4% for each equipment with a corresponding probability of material discard or forward process (material not discarded). Setting the probability of failure to greater than zero means that as the batches(or lots) enter the Run subactivity of each activity, that block will process a failure based on the probability. These blocks have been specifically designed to deal with failure and are called 'Subactivity w/Failure' blocks which means that the failure rate is always present. Entering a zero simply switches it off and therefore the control for this parameter is within the Excel input file.

In order to run the sensitivity analysis a number of assumptions were made, as listed below. The input parameter values used can be found in Table 5.6.

- Single product/campaign
- Batch demand set to 55
- Column parameters only of Capture/Protein A
- Split ratio limit of 8
- Fixed downstream batch split/combine locations
- Failure rate of zero unless being tested

Table 5.6 Summary conditions set for sensitivity analysis for all titres

Parameter	Base Value	Change
Titre	1.5g/L, 4.5g/L or 10g/L	±20%
Resin Lifetime	80 cycles	±20%
Yield	Equipment dependent	±5%
Capture DBC	20g/L	±50%
Failure Rate	4%	±50%
Column Height	25cm	±8%
Column Diameter	60cm	±20
Split Ratio Limit	8	±20%

As stated earlier, the model was already set up to deal with any probability of failure or failure rate. Similarly, the other parameters could easily be changed by different inputs in the Excel input file. For example, if the resin lifetime was changed from 80 to 90 cycles, the 'Repack Check' block would read this value and send the column to be repacked every 90 cycles instead. Similarly the yield input would be automatically read by the 'Mass Balancing' block and accounted for in the calculations. Column dimensions, DBC and the split ratio limit were used for the calculation of the split/combine ratios which occurred within the Excel input file. The model would not be aware of any changes to these parameters other than the number of cycles through the columns which may have differed. Similarly, titre was used for the calculation of the split ratio and also to determine the number of batches required to meet a kg demand. Within the model the number of batches created for each campaign would be based on this titre, however again, the calculations would have all been carried out in the Excel input file.

5.4.1.2 Deterministic analysis results

Using the parameters stated under Table 5.6, the model was run deterministically and the kg throughput of each run was recorded. Using the base case (0% variability for all parameters) the impact of each parameter change could then be recorded as a % change in kg throughput against the base i.e. against 0%. Figure 5.6 shows the results of the analysis for all three titres using Tornado diagrams.

Figure 5.6a shows that at the lowest titre the biggest impacts are due to the step yields, failure rate and titre. The column height, split ratio limit and resin lifetime, have no impact and the column diameter and DBC only have negative impact. This is due to the fact that at low titres, the number of splits or cycles required through the Protein A column is 6 and therefore well below the limit. Increasing the column dimensions or the split ratio limit will not make a difference to the amount of product able to bind as it is already maximised. Likewise, decreasing the split ratio limit by 20% sets it to 6 and therefore still allows for the required cycles. Decreasing the column diameter and DBC however raises the required cycles to 10 and 12 respectively thus resulting in a decrease in the amount of product actually binding. The overall kg throughput is therefore reduced.

Conversely, at the higher titre of 4.5 g/L, the base split ratio limit is exceeded with 18 cycles required to process all of the product. Therefore the changes in yield and titre become less and less significant and the factors affecting the column capacity begin to dominate i.e. DBC, Diameter, and Split ratio limit as illustrated in Figure 5.6b.

A similar trend can be seen in Figure 5.6c at the 10g/L titre. This suggests the existence of a downstream bottleneck at higher titres, most likely the capacity of the capture column. It must be noted that the resin lifetime does not have an impact on kg throughput at any of the titres. This is due to the fact that the number of cycles is limited to 8 regardless of split ratio requirements. The maximum number of cycles that the capture column performs is therefore never greater than eight times the number of batches. If the split ratio limit were removed this parameter would have far greater impact as repack would be needed more often.

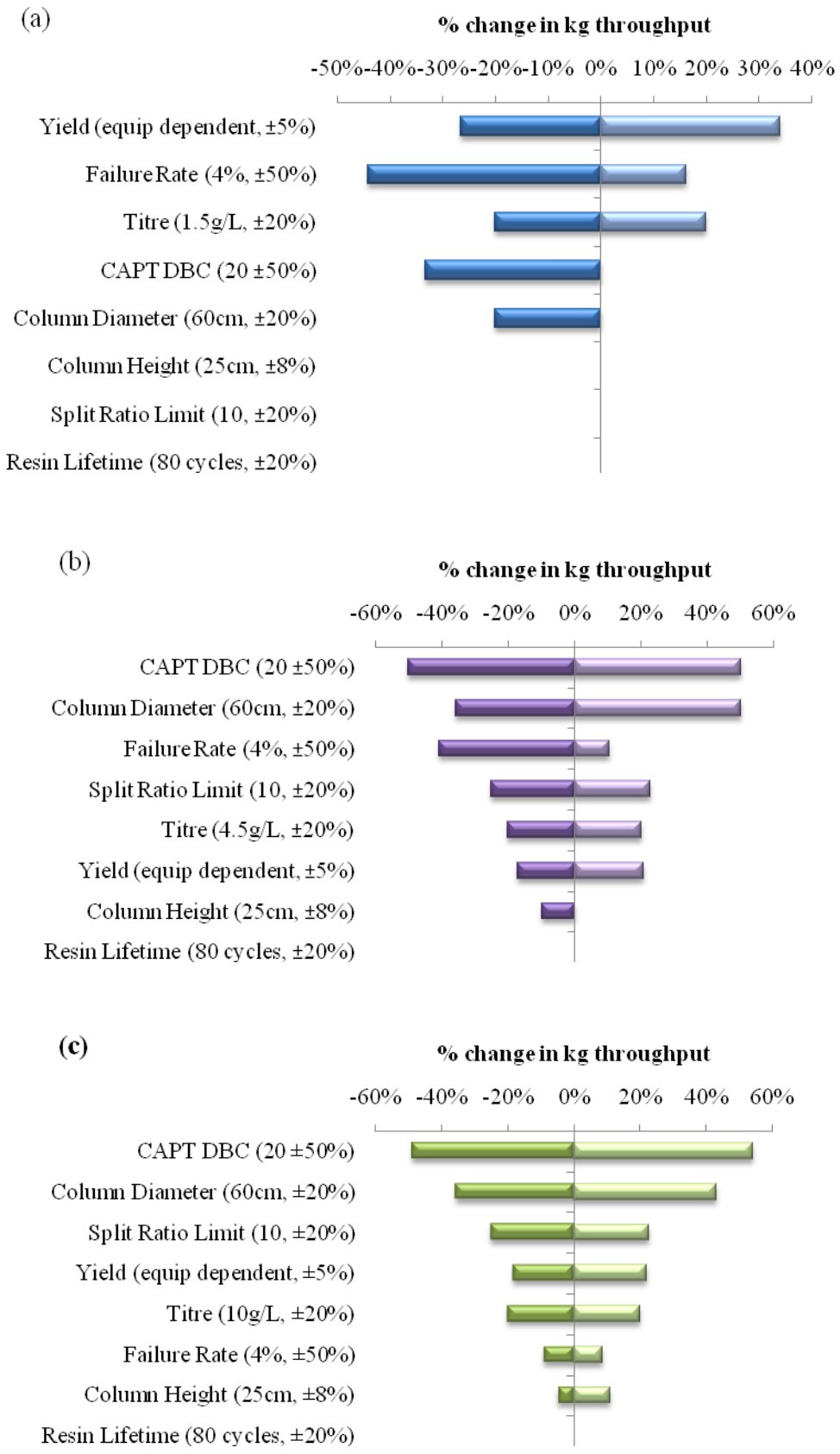


Figure 5.6 Tornado diagrams showing the sensitivity of the kg throughput to keyinput variables at difference titres: for (a) 1.5 g/L, (b) 4.5g/L, (c) 10g/L

5.4.2 Scenarios Analysis

As a result of the sensitivity analysis a methodology has been formulated for scenarios analysis, looking at different possible strategies to deal with the identified downstream bottleneck. Figure 5.7 is a flowchart illustrating this methodology.

5.4.2.1 Scenario 1 analysis setup

The deterministic analysis showed that the process is sensitive to the capture column parameters, particularly as the titre increases. Since the capacity of the downstream columns, in particular the capture column, is the limiting factor, then changes in column parameters should be considered as a strategy for debottlenecking. Scenario 1 therefore asks the following question:

Given fluctuations in current and future titres, what strategy should be adopted in order to minimise resin costs while maximising % process throughput (kg) and achieving a reasonable split ratio. The strategies are as follows:

a) Buy a new capture column (diameter change)
b) Buy a new Protein A resin (DBC)
c) Increase capture column height (height change)
d) Improve process efficiency (failure rate)
e) A combination of the above

These strategies were chosen as the options because the parameters corresponding to them were found during the deterministic analysis to be highly impacting on the process throughout.

Table 5.7 Summary of input parameter values for scenario1

Titre (g/L)	DBC (g/L)	Diameter (cm)	Height (cm)
1.5 ±20%	20	60	25
4.5 ±20%	30	70	27
10 ±20%	50	80	
		90	
		100	

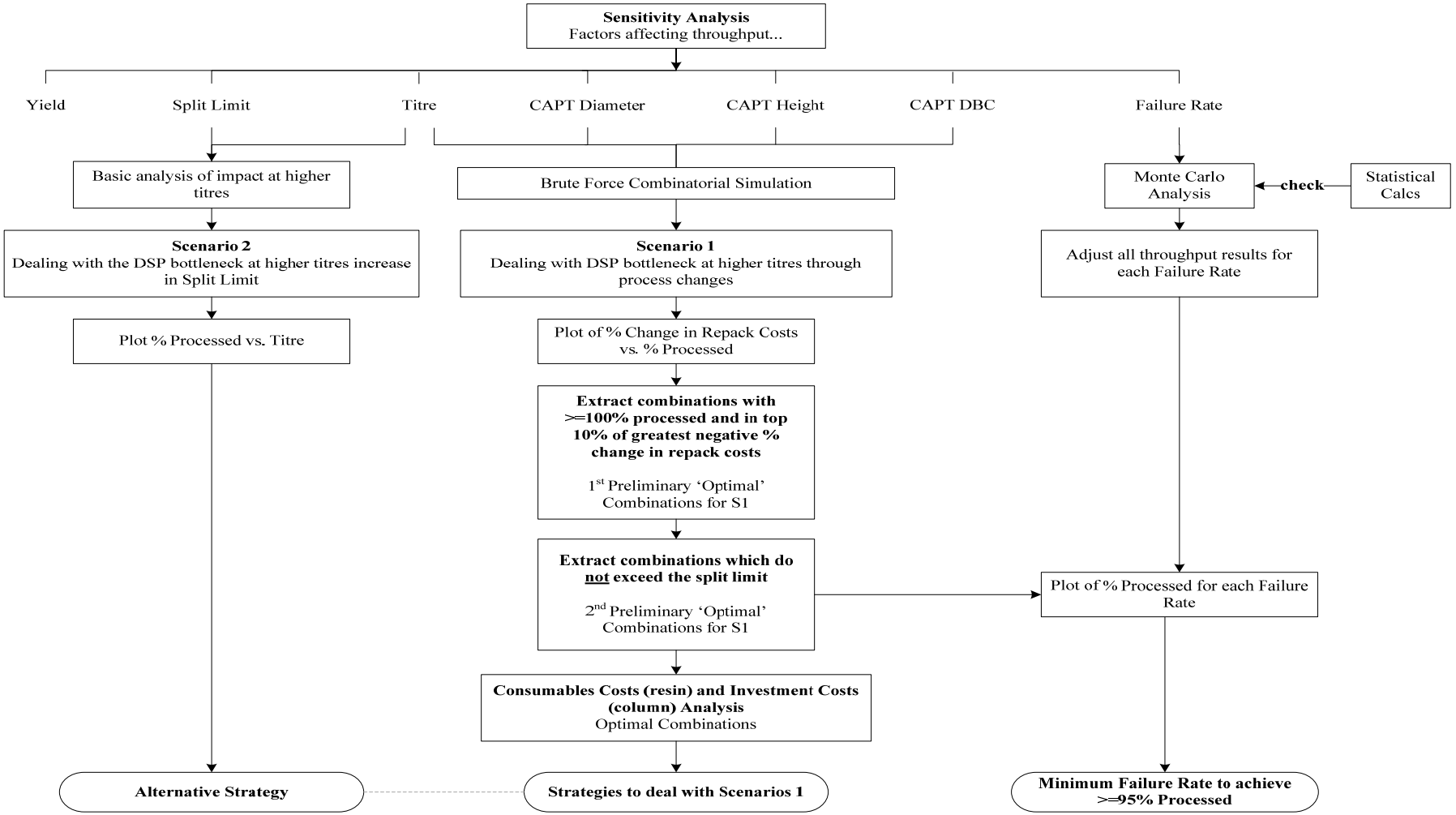


Figure 5.7 Flowchart showing methodology for scenarios analysis

Titre:The manufacturing company have stipulated that their process has been validated for ±20% fluctuations in titre. It is therefore assumed that this is the general trend which they have observed and can be used here for each of the main titre inputs as a typical trend.

DBC: Taking the GE Healthcare ProteinA resin MAbSelect as an example of a typical resin used, it can be assumed that there is a range of dynamic binding capacity achievable between 20-30g product/L. The model uses a DBC value of 20g product/L a value typical for current ProteinA resins and the lower DBC achieved with MAbSelect. The upper range of 30g/L is set as a value achievable without any changes to the resin. Furthermore, any split ratios determined for this resin take the upper range value for their calculation, a technique adopted to maximise load efficiency. There is also a newer GE resin called MAbSelectXtra which has a binding capacity of 50g /L.

Height: The recommended mobile phase velocity for the MAbSelect resin is 500cm/hr however most common velocities are stated to be around 300cm/hr for most resins (and therefore the assumed base case value). Although the higher the height, the lower the velocity (and higher the residence time), it is assumed that a 2cm increase in height to 27cm is allowable within the operating range of the resin.

Diameter: Any change in diameter from the base case will mean purchasing a new column. MAbSelect and MAbSelectXtra have available for them variable size columns which reach 120-150cm in diameter.

There are thirty different combinations of parameters as shown in Table 5.8. Given the ±20% fluctuations there are 90 combinations for each titre value, giving a total of 270 simulations run.

Table 5.8 Different combinations of parameters used in scenario 1

Combination	Capture DBC	Capture Diameter	Capture Height
1-30	20	60, 70, 80, 90, 100	25, 27
31-60	30	60, 70, 80, 90, 100	25, 27
61-90	50	60, 70, 80, 90, 100	25, 27

5.4.2.2 Scenario 1 analysis results

For each combination the kg throughput has been recorded and given as a % processed based on the required to actual throughput ratio. This is plotted against the 'cost of repack', the cost incurred every time a column is repacked with a particular resin. It must be noted that the initial purchase cost for the resin has not been taken into account due to the fact that these costs are used for comparison analyses and therefore only relative numbers are necessary. Figure 5.7 shows these plots.

As figure 5.8a shows, at the titre of 1.5g/L the number of cycles required to fully bind 100% of the product is 6, below the set maximum split ratio limit of 8. This means that the % processed at the base case is already above 100% (greater than 100% is possible because the batch demand overcompensates for step losses). Any changes in column dimensions will have an effect on the cost of repack as the volume changes however the % processed will remain constant. Therefore optimal combination selection at this titre will not be based on meeting the split ratio limit or the greatest % processed achieved but any reductions in cost of repack.
Figure 5.8b shows that as the titre increases to 4.5g/L it can be seen that the base case lies just below 50% processed. The reason for this is that 18 cycles are required to fully bind the product and therefore the split ratio limit of 8 is simply not sufficient. In fact, when compared to the previous graph for 1.5g/L a definite shift can be seen as fewer combinations result in 100% processed. Those which fail to meet this value, unsurprisingly, also exceed the split ratio limit in terms of split requirements.
Figure 5.8cfor 10g/L shows a dramatic shift along the x axis with very few combinations actually resulting in 100% processed. This shift is expected; at this titre 40 cycles are required for fully bound product. Only extreme process changes will compensate for this high cycle requirement such as a large increase in binding capacity and column diameter.

From these graphs, combinations for all titres can be found which are deemed to be preliminary 'optimal' based on the following criteria:

- Meeting the split ratio limit
- Achieving greatest % process throughput
- Greatest negative % change in cost of repack against the base case (taking only the top 10 percentile)

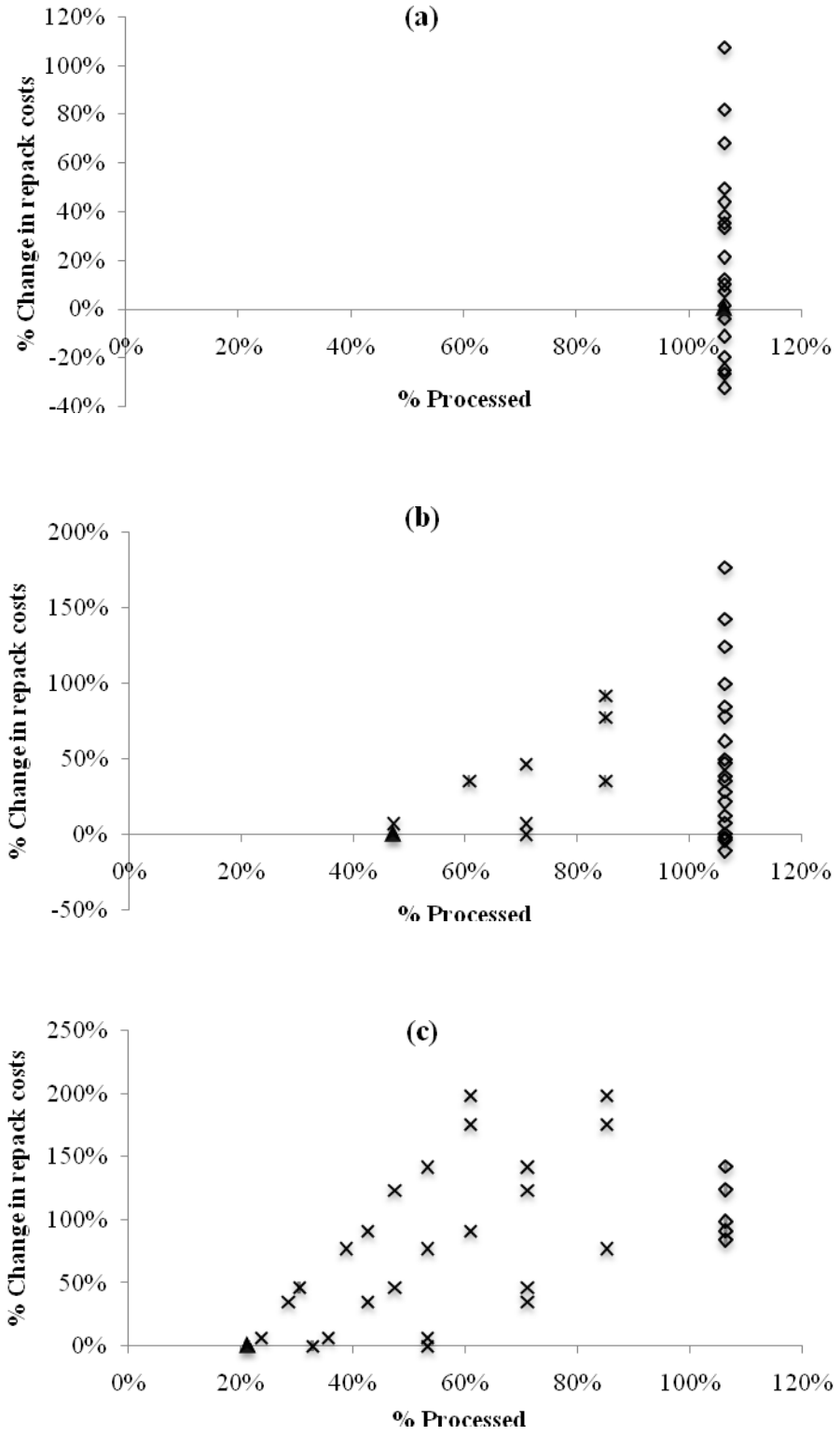


Figure 5.8 % Change in cost of repack against % processed for different combinations of Capture column DBC, diameter and height at (a) 1.5g/L, (b) 4.5g/L, (c) 10g/L titre. Where (▲) is the base case, (◊) is below split ratio limit and (x) is above split ratio limit.

These combinations are deemed 'preliminary optimal' because they do not take into account the investment costs which will be incurred if the process changes are made. It must be noted that these investment costs only account for the bare cost of purchasing a new column and not the cost of purchasing a new resin, nor the costs associated with personnel training or validation. Thus, taking these preliminary combinations the investments costs have been factored in, using a combination of the Cost of Repack and the amortized cost of investment to give an annual cost for each combination. Ordinarily amortization is applied to intangible assets such as patents and loans and depreciation is used for tangible assets such as equipment. However it is being assumed here that the columns have zero scrap value which is the only element differentiating depreciation and amortization. Thus the latter can be used. Furthermore, it is also being assumed that the equipment lifetime is 10 years, which is the estimated lifetime of a bio facility.

Based on the annual costs which combine the cost of repack and the amortized cost of investment, the secondary optimal combinations can be found (these are in fact the final optimal combinations). Thus, looking at the combination with the lowest annual cost, the optimal combinations give the following strategic solutions for each titre.

At 1.5g/L, 100% process throughput is achievable without any process changes. However, in order to save on cost of consumables (around 30% reduction in annual cost of resin), the following change can be made: resin change from MAbSelect to MABSelectXtra to achieve the higher DBC of 50g product/L resin and capture column diameter increased to 70cm (new column purchase)

At 4.5g/L, process changes will have to be made in order to achieve 100% process throughput. These are: resin change from MAbSelect to MABSelectXtra to achieve the higher DBC of 50g product/L resin, capture column diameter increased to 80cm (new column purchase).

At 10g/L, in order to achieve 100% process throughput, the process changes are: resin change from MAbSelect to MABSelectXtra to achieve the higher DBC of 50g product/L resin and capture column diameter increased to 100cm (new column purchase)

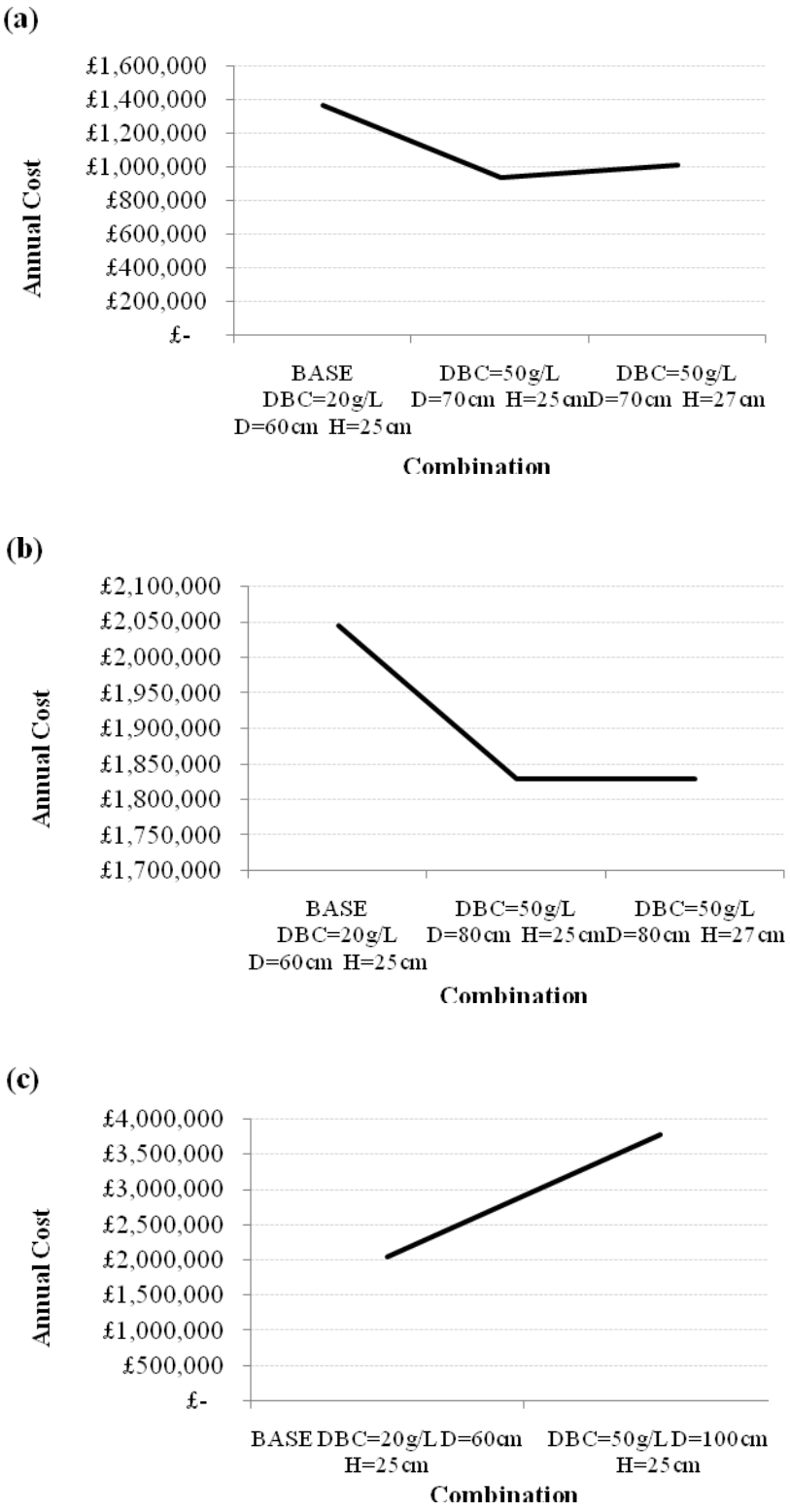


Figure 5.9 Annual cost for preliminary optimal combinations factoring in amortized cost of investment based on equipment lifetime of 10 years and cost of repack found at (a) 1.5g/L (b) 4.5g/L (c) 10g/L

5.4.2.3 Failure Rate

The deterministic analysis found that the failure rate is a significant parameter in the mAb process and that a variation of only a few percent has a large impact on the process throughput.

Since failure rate is based on probability it was not included in the combinations analysis, simply because the failure would differ with each run. The stochastic nature of the parameter calls for a stochastic analysis, in this case Monte Carlo simulations, in order to obtain a figure for process throughput with a great enough convergence to deem it accurate. Using four different failure rates: 1%, 2%, 3% and 4% the Monte Carlo simulations were run with a demand of 50 batches. The number of batches processed were recorded with the following results as shown in Figure 5.10.

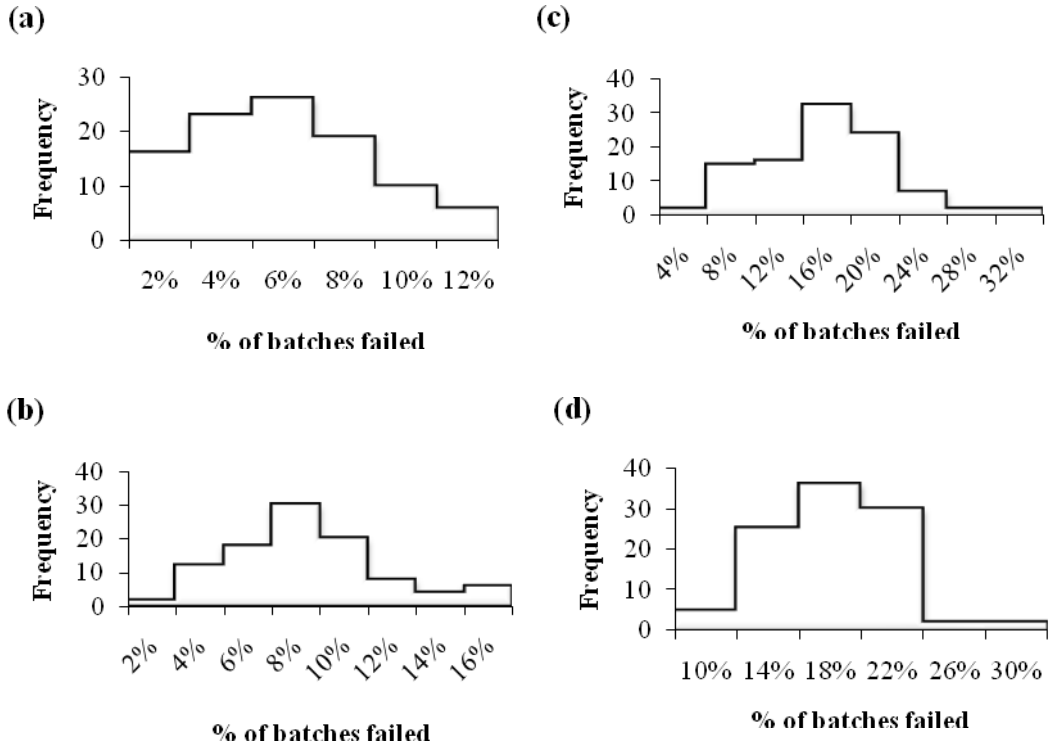
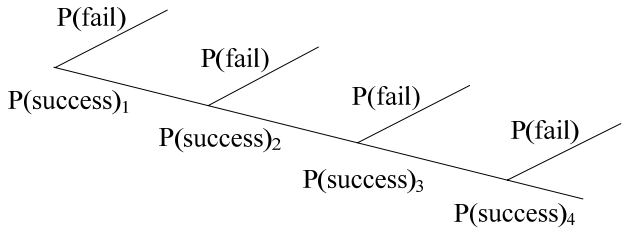


Figure 5.10 Distributions for % batches failed for (a) 1% (b) 2% (c) 3% (d) 4% failure rates

According to the highest peaks the % of batches failed out of the possible 50 is 6%. 8%, 16% and 18% for the different failure rates respectively. These figures however need validating and this was done by doing simple statistical calculations.

Taking a simple statistical approach, the probability of failure here can be calculated by taking the unit operations in series which are affected by failure rate and multiplying their corresponding probabilities of failure to get the overall impact. In order to ascertain the number of batches out of the possible 50 which are successfully processed, it is necessary to consider the probability of success rather than failure, where P(Success) is the probability of success. For example, for a failure rate of 1%, the probability of success is 99% (100%-1%). Since there are four units in series which a batch must successfully pass through, the probability of success is powered to the number 4. This gives a figure of 0.96 or 96% success rate. Multiplied by a possible 50 batches, this gives a 48 batch throughput. As shown below, the same calculation can be applied to the different failure (or success rates) to determine the expected batch throughput.



P(success) for a batch = $P(success)_1$ x $P(success)_2$ x $P(success)_3$ x $P(success)_4$

1% Failure Rate: P(success) = 0.99^4 = 48 Batches or 2 fail (4%) 96%
2% Failure Rate: P(success) = 0.98^4 = 46 Batches or 3 fail (8%) 92%
3% Failure Rate: P(success) = 0.97^4 = 44 Batches or 6 fail (11%) 89%
4% Failure Rate: P(success) = 0.96^4 = 42 Batches or 8 fail (15%) 85%

The calculated figures are a few % lower than those obtained through the Monte Carlo analysis. This discrepancy however is expected due to the fact that if 5 out of 50 batches fail, this does not mean that remaining 45 will be processed. Each failure has a TTR or Time to Repair associated with it which reduces the overall time available to process the batches. This consequently drives up the % of batches failed

because it also accounts for the batches that did not have time to get through but did not actually fail. But as the calculated values don't take this into account, the % failed is lower.

As the figures are only a few percent different, the results of the Monte Carlo analysis can be used, taking the % batches failed to adjust the outputs from the combination analysis. Taking only those combinations deemed optimal at the preliminary stage the % processed has been plotted for all four failure rates by adjusting the kg throughput (assuming that % failure of batches can also be applied to kg throughput). Figure 5.11 shows the plot for 1.5g/L.

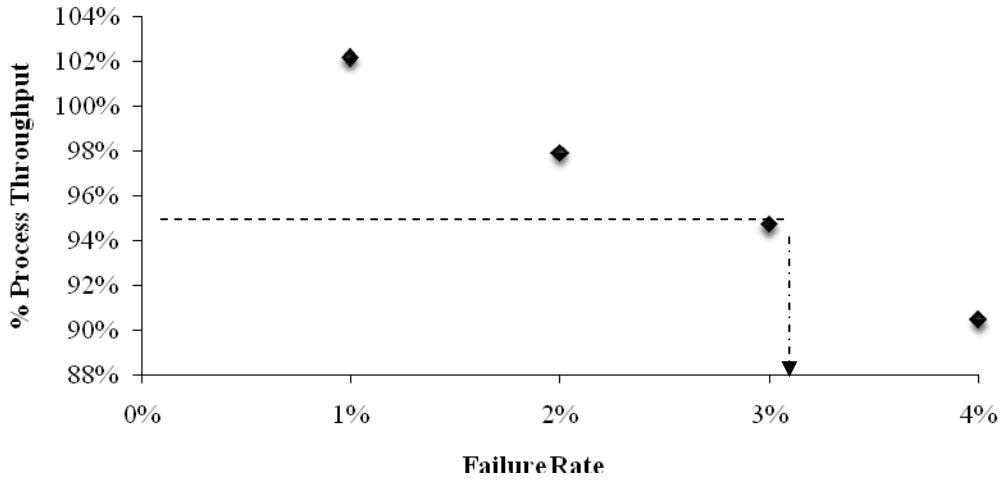


Figure 5.11 Adjusted % process throughput, for different failure rates, for combinations with high relative % decrease in repack costs for 1.5g/L. Arrow illustrates failure rate target to achieve 95% process throughput

Although not shown, the plots for the 4.5g/L and 10g/L titres showed identical trends. Assuming that it is desirable to achieve a minimum of 95% processed, the plot shows that the failure rate must be below 3% for all titres.

5.4.2.4 Scenario 2 analysis setup

In order to understand the extent to which the downstream bottleneck is linked to the split ratio limit a quick analysis has been performed which measures the amount of product processed at each of the titres, relative to the amount of product which

should have been processed. If it is assumed that the facility is used to full capacity and therefore the number of batches required remains constant, then a certain kg demand/required is set based on titre, production fermenter volume and process yield. This means that if a percentage of product fails to bind, the overall process throughput drops as a percentage of the kg demand. Thus, the percentage actually processed can be determined using the following ratio:

$$\% \text{ Processed} = \frac{\text{Kg Throughput}}{\text{Kg Required}} \qquad (5.4)$$

Plotting the % processed and the split ratio required against the different titres, Figure 5.12 shows the % processed decreases dramatically with each titre. This is explained by the solid line representing the required split ratio which far exceeds the split ratio limit as titre increases. With constant column dimensions the capacity of the column to bind the product remains the same while demand for binding capacity increases. A large amount of product therefore simply flows through.

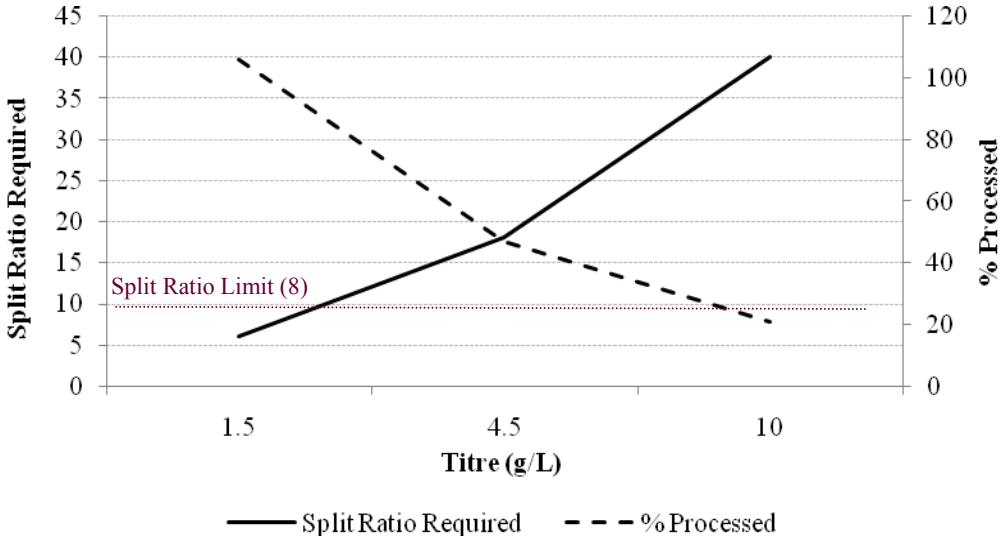


Figure 5.12 Split Ratio required and % processed for each titre assuming constant batch demand and removal of the split ratio limit of 8.

This suggests that if the split ratio limit is increased, the % processed should also increase. In order to examine this theory, the process will be run for the three different titres, with different split ratio limits ranging from 10 to 18.

5.4.2.5 Scenario 2 analysis results

Figure 5.13 shows the % processed for the three different titres, with the different lines corresponding to different imposed split ratio limits.

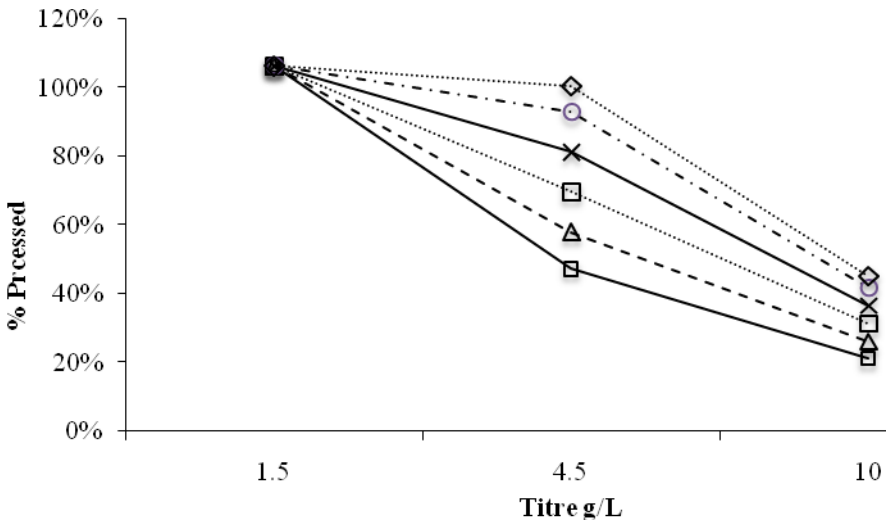


Figure 5.13 % Processed with different split ratio limits for different titres, assuming constant batch demand. Where (—□—) is split ratio limit of 8, (-△--) is split ratio limit of 10, (···□···) is split ratio limit of 12, (—×— is split ratio limit of 14, (-·○·-) is split ratio limit of 16 and, (···◇···) is split ratio limit of 18.

As the number of cycles allowed increases, the % processed increases also. When the limit equals the required split, % processed jumps to 100%. At 4.5g/L, this is 18. At 10g/L this would be 40.

Therefore, an alternative to adopting the strategy proposed in Scenario1 is to increase the split ratio limit. However it is not really feasible at higher titres due to the sheer number of cycles required.

5.4.3 Evaluation of use of the Standard Framework in Construction of mAb

The mAb model was based on two previous models built during the design stages of the process. The functionality of the new model compliments the new scope which has various overlaps with the scopes of the original models. In order to understand and evaluate the value of using the Standards Version 2 framework to construct the mAb model a qualitative analysis must take place. In an ideal world, a quantitative analysis would also be performed however it is exceedingly difficult to quantify the comparison between the process of two construction methods. What can be offered is the feedback from the end users and the validation outcomes of data/trend verification, using the requirement specifications as a guide.

Intuitive to user

Under Chapter 3 it was discussed that in order for a model to be intuitive to the user, certain guidelines can be followed, as given by Valentin and Verbraceck (2002). These are that the 'Interactions between model parts...should represent interactions in the real system', 'Use concepts that represent functionalities as found in reality and that can be used for visualisation purposes' and 'Visualise a system in such a way that complexity is reduced but the essential processes are still shown'. The latter point simply ensures that the level of detail built into the model, both in terms of coding and blocks, is sufficient to meet the immediate scope and likely additions in the future (in order to maximise reusability) but no more. This works on two levels; firstly, limiting the level of detail to the necessary amount limits the resulting model run time and secondly, the risk of confusing the user is minimised if they can associate the level of detail with the well defined model scope. The first two points ensure that the visual parts of the model represent the real system so that the user can easily associate model elements to the system elements. In the case of mAb the user does not see the actual model itself (unless structural changes are being made) and only the input/output Excel files are seen. These however, as stated under the next section, have been designed to maximise user intuitiveness.

Relevance and Ease of Data Input/Output

In order to enter data into the database, the original models used the SDI link offered by the Extend platform. This meant that: the database source within Excel had to

have the SDI structure, the Excel database would have to be manually exported and imported in order for any changes to be applied in the model and the Extend file would have to be accessed in order to achieve this.

The new model however has been built such that the Extend file need never be seen by the end user unless they need to make structural changes to the model. The input file or the source database has a user-friendly structure which industrialists have indicated is in the format that they are used to seeing. Using a macro the Extend model automatically imports the data it needs, runs and then outputs the results into a different Excel file where the data is already set up to be in a user friendly format, with Gantt charts and tables of data summarising the process output. This method of construction means that an end user can easily vary any input parameter (all parameters have been set to be variable) and needs not have any Extend experience.

Maximised reusability and sustainability

The inputs and outputs to the mAb model are quite extensive, covering labour/operating shifts, utilities usage, multiproduct changeover and mass balancing. The number of fermenters can also be changed by inputting the number desired, although greater than four 5k fermenters will require a structural change. Currently the facility is set to be used for one product only however a number of monoclonal antibody products could be manufactured using the same platform process. The set up of the model accommodates for the introduction of these new products, but with a maximum of three different products produced in the facility during any one year or model time horizon. This limit reflects the typical number of products expected in a commercial facility (personal communication, Roger L Scott, Eli Lilly, Indianapolis)

Also, the use of the mAb model to perform the deterministic and scenarios analysis is a further testament to its capabilities. In this chapter is has been shown that the robustness of the system can be tested through a deterministic analysis, very simply carried out by changing the input parameters made readily available to the user. The scenarios analyses have also shown that more in-depth analysis can be made, making the model a useful decisional tool in strategic analysis based on future uncertainties. The following chapter will go on to illustrate the models capabilities in a multiproduct setting.

5.5 Conclusion

Much like the BioSynT case, the mAb case study was used to demonstrate the ability of a model, built using the standard version 2, of being used as a decisional tool. A deterministic study was first carried out in order to determine those parameters whose variability would significantly impact the output metric, kg throughput. It showed that at different titres, different parameters had impact, with the capture column parameters becoming more significant as the titre increased.

A stochastic study was then carried out using the results of the deterministic analysis, looking at the impact of different scenarios combined with process uncertainties to determine how these would affect the running of the process. Combining these with a cost analysis of various process options it was found that at all titres it would be beneficial to upgrade the resin used for the capture step in order to gain a higher dynamic binding capacity and that a new column (for increased height) would give greater process binding efficiency.

Furthermore, in order to achieve a % process throughput of greater than 95%, the failure rate would have to remain below 3% for all titres.

The study also found that as the number of cycles allowed increases, the % processed increases also. When the limit equals the required split, % Processed jumps to 100%. At 4.5g/L, this is 18. At 10g/L this would be 40. Therefore, an alternative to making expensive column changes would be to increase the split ratio limit.

Finally the mAb case study was used to implement the Standard Framework 2, testing its ability to guide the construction of a manufacturing capacity management model capable of meeting the requirement specifications stated under the standard. A qualitative analysis showed that these were met to a far greater degree than illustrated with the BioSynT model, with significant improvements in model development time, user intuitiveness, reusability and sustainability, and ease of data input/output. Some shortcomings were also identified however (as discussed in Chapter 3) which necessitate further development of the standard framework.

CHAPTER 6

Multi-Product Campaigning: Schedule Optimisation

6.1 Introduction

The increase in the number of drugs in manufacture and their large volumes has meant an increase in the number of multi-product facilities of which, one may consider, there being two kinds. The first being whereby different products are made in parallel, simultaneously and in different production suites, and the other whereby different products are made on a campaign basis in the same part of the production facility.

An example of the former is the Bayer Multipurpose Biotechnology Plant, a pharmaceutical manufacturing facility in the US, which currently produces three recombinant protein technologies. Over half of the plant is dedicated to the manufacturing suites with each suite producing a separate product without interrupting other processes. Not all pharmaceutical companies however are able to have such large in-house production capacity and currently 35% of biomanufacturers outsource at least some of their biologics with these manufacturers projecting that by 2008 47% will outsource at least some production (Langer 2004). As more and more companies are employing this strategy, contractors are having to cope with the varying operations due to the variations in product demands and requirements. Some plants may use dozens of equipment to produce several different types of products leading to a myriad of ways in which a plant can be operated and finding the best operating plan and schedule has become a challenge. In order to optimise productivity in multi-product, contract-manufacturing facilities where processes are

run in parallel requires a large degree of overlap between batches and effective utilisation of shared resources and equipment (Gosling 2003). The correct scheduling of such processes is therefore vital to the production optimisation of such facilities.

There are certain cost implications associated with scheduling and unit utilisation. Multiproduct facilities are faced with the problem of having to maximise productivity and the effective utilisation of resources while reducing the costs associated with such production activities. As with all projects, more often than not, the optimisation of performance will mean a compromise between time and cost. If you want to fulfil full production capacity in the shortest and most optimal amount of time then that will more than likely mean spending more money.

As manufacturers shift from a production driven focus to a demand driven one, manufacturing performance becomes increasingly important. Effective planning and scheduling is one of the keys of meeting manufacturing goals (Taylor). In biopharmaceuticals it is vital that product demands are met with sufficient supplies with certain products given priority over others depending on the circumstances. Therefore firstly manufacturers will be looking to optimise the scheduling and secondly at reducing the costs of running such multi-product facilities.

6.2 Operating Strategies in Biopharmaceutical Multi-product Manufacture

If the production of monoclonal antibodies occurs in various clean rooms then when a product changeover occurs in the facility, as part of a multiproduct campaign sequence, these rooms must be shutdown and cleaned before the next product may enter. The procedure is known as ‘turnaround’. This ensures that cross contamination is minimised between the products and can also serve as a maintenance downtime. Since these rooms are sealed it is possible to validate the process such that different products can be processing in different rooms at the same time. This opens the way for two different types of product changeover. The first, called Synchronised, means that each campaign/product must finish processing in the facility i.e. across all rooms before the next campaign can enter. This means that if the first room has finished the

last batch of a campaign, it will not start its turnaround procedures until the last room has also processed that batch. Only then will all rooms be turned around at the same time. There are disadvantages to adopting this procedure, for example, if all rooms are to be turned around at the same time, the labour may become a constraint. Also the process throughput may be affected because the cycle time of each campaign increases. The second method of turnaround is called 'Rolling' where rooms need not wait for the entire process to finish with a campaign i.e. as soon as a campaign has left the first room, the turnaround procedures may begin. The implications of this method are that if labour requests for room turnaround are one after the other, labour may not be a constraint. Also, the cycle time of each campaign is reduced meaning that theoretically the process throughput could be increased.

In addition to the procedures surrounding multi-production manufacture, a common variability among different manufacturing systems is the operating shift adopted. That is, the start time and duration of the labour shift. Depending on the degree of automation in a process, the labour shift will have an impact on the process scheduling. Below are some examples of rules which have an effect on how a process is run and which therefore must be captured in a model for true system representation.

- An activity can only begin if it will finish running during the current shift
- An activity can only occur during day/night shift
- Certain sub-activity sequences must be processed in one go

A shift pattern, as used in the context of this case study, is defined as the hours during which labour is available and can comprise of more than one labour shift. For example, if there are two labour shifts, one can run from 6am to 6pm, the other from 4pm to 12am. There is a period of overlap between the shifts but ultimately there is labour coverage from 6am to 12am, making it an 18hour shift pattern.

If a shift pattern associated with a process is 24hrs 7day this means that theoretically the process can continue non-stop until a scheduled shutdown or product changeover. The only delays will be due to various other constraints such as resource availability and such rules as those outlined above which are associated with the shifts that make up the shift pattern. If however, the shift pattern is 18hrs 7day, and assuming that the process does not continue without labour shift coverage, these constraints have an

effect as well as the forced stoppage of 6 hours. The cascading effect of these stoppages on the overall schedule and the achievable process throughput can be quite significant.

6.3 Case study background

Chapter 5 illustrated an example of using the standard framework to create a biotechnology manufacturing model capable of single product capacity management scenario analysis. It demonstrated such features in the model as parameter variability, split and combine ratios and batch failure. The constructed model however contains various other interesting features which have not yet been explored such as, multiproduct campaigning, product changeover procedures and operating shifts.

A case study examining the use of these features will now be presented, looking at the optimal campaigning schedule of multiple products, given a set of constraints, according to three quantitative measures.

6.4 Method

Section 6.2 discussed how the process elements, Turnaround strategy and Shift Pattern can affect the scheduling of a process and thus the achievable throughput. This section will discuss how these elements have been captured in the mAb tool.

6.4.1 Turnaround

There were many challenges faced with regards to turnaround. Firstly the synchronised approach would mean that all process areas would have to be turned around at the same time however the flow of batches would have to be controlled vigorously so as to ensure that no activities were still ongoing when turnaround began. To tackle this, the model captured the turnaround procedure using a block which looked identical for all rooms. This block sat before all blocks within each room and every time a batch entered, it checked to see if it was the last batch of a campaign. If not then the batch simply went on to be processed. If however it was

recognised as the last batch then turnaround was triggered by splitting the batch iteminto two parts, the first representing the original full batch which went on to be processed, the second the turnaround item which would wait until that batch had been processed. If synchronised turnaround had been set, then the campaign gate would be shut, not allowing the next campaign through to begin processing. Once each room had finished processing the last batch of the current campaign, the global array holding the activity status of each process area would indicate a 'ready'. Once all areas had been set to ready, the gate for turnaround would open and allow the procedure to go ahead. Since each room would have its own turnaround item generated as the batch went through, these items would go on to the turnaround activities which would call upon the required labour and would execute turnaround procedures with the predefined cycle times. The second challenge faced was the possible synchronisation of turnaround with area shutdown which meant that if turnaround was needed and shutdown was required within the following two weeks, shutdown would be brought forward to occur at the same time, thus reducing the period of downtime. To capture this, the turnaround block contained logic to decide whether shutdown was required.If so, both a turnaround and a shutdown item would be generated to process both activities. The entire campaign changeover procedure was therefore automated meaning that the model could trigger turnaround according to whatever campaign schedule the user input.

6.4.2 Operating Shifts

This feature was modelled with the assumption that only two different shifts would be required. These shifts could be any length, start at any time, apply to any number of days per week, have any number of labourers allocated and could overlap. The user would put this information in the input file and the model would apply it to the labour resource pool by making labour available (or unavailable) according to the shift specified. The shift pattern specified had an effect on the process throughput because activities/sub-activities cannot process unless the required labour is available. For example a 24hr 7day shift would allow for continuous processing and would therefore most likely result in a higher process throughput capability than a 18hr 5day shift pattern.

6.5 Results and Discussion

The following section will describe the setup of a multi-product scenario looking at campaign scheduling, and the results achieved through basic method optimisation.

6.5.1 Scenario Analysis Setup

The scenario question was as follows: given three different products with the characteristics shown in table 6.1, if a certain number of batches of each product are required over a one year time horizon, what is the 'optimal' campaigning sequence given different turnaround strategies and operating shifts? Where the different turnaround strategies are Synchronised or Rolling and the different operating shifts are 18hr7day and 24hr7day.

Table 6.1 Summary of product characteristics

Product	**Description**	**Titre (g/L)**	**Demand (Batches)**	**Demand (kg)**
A	Manufacture	1.5	40	203
B	Validation/consistency	2	5	34
C	Phase III Trials	3.5	2	24

In order to model demand, the time horizon for the scenario will be one year, with the year split into four quarters. Demand is defined as the amount of product needed where the deadline is given as a quarter.

Constraints

- Validation and trial batches were needed as early on as possible in order to speed up approval
- A certain amount of the manufacture product A needed to be made first in order to meet demand while the facility was being used for products B & C.

Assumptions

The following assumptions were made in the running of the scenario:

- Validation batches were needed by the end of the first quarter
- Trials batches were needed by the end of the second quarter
- At least half of the manufacture batches must be made by end of second quarter
- All batches for validation and trial products must be made in one campaign but production of the manufacture product could be split
- Product A would be split into a maximum of two campaigns of even sizes
- There were no constraints on the order of campaigns
- There was no storage limit for the manufacture product
- The 'ideal' campaign strategy would minimise plant downtime
- Buffer and labour requirements of product A = product B = product C
- The facility/process had a zero failure rate therefore no batches or lots were contaminated and/or discarded
- In synchronised mode, enough teams were available for turning around all rooms

Demand

The following demand profile was used, constructed according to the assumptions and constraints outlined above. The deadlines were for end of quarter for example, if five batches of B were required in Q1 then the deadline would be at the end of Q1.

Table 6.2 Product demand profile

Product	**Q1**	**Q2**	**Q3**	**Q4**
A		20		20
B	5			
C		2		

Metrics

Customer service level: defined here as the (batches produced/batches demanded) for each time period. If all batches meet demand within a time period then the CSL value for that quarter would be 1. If none were produced in that quarter, then the CSL

value would be 0. If a product was not produced at all during the time horizon, then the CSL for that product would be -1. This ensured that there was a quantitative difference between a product being produced late and not being produced at all.
Cost: defined as [cleaning cost per day after a campaign + start-up cost for a campaign]. The actual cost values to used were arbitrary as only relative costs were needed for campaign comparison. The desired outcome would be as low a cost of changeover as possible.

OEE: This metric was used here to capture the downtime due to the campaign changeovers as well as the scheduled facility shutdowns. It was defined as:

$$\text{Quality x Availability x Performance} \tag{6.1}$$

where:

Quality = 'good' batches produced/total batches produced including contaminations. Since it was being assumed that the process has 0% failure rate, the Quality parameter is 100%

Availability = Operating time / Planned Production Time

where:
Planned Production Time = horizon length – scheduled shutdowns
Operating Time = planned production time – downtime due to turnaround

Performance = often called Product Rate, was defined as the average output from the process divided by the proven maximum output over the same time period. Usually a one week period is chosen as the timeframe. However, in this case, performance was taken to be the equivalent of the CSL value, as the latter effectively measured the performance of the campaigning sequence against the expected or demanded value.

Campaign combinations

A	B	C	
A	C	B	
B	A	C	
B	C	A	
C	A	B	
C	B	A	
A	B	A	C
A	B	C	A
A	C	A	B
A	C	B	A
B	A	C	A
C	A	B	A

Model outputs

In order to record the metrics the following outputs were needed:

- Completion date of each campaign
- Number of batches processed in each campaign
- Number of turnarounds and durations

6.5.2 Scenario analysis results

In Appendix C the results of the different campaign sequences can be found. These tables show, for the different turnaround strategies and shift patterns, the throughput results of each simulation run for each sequence and the calculated Cost, OEE and CSL values based on the number of turnarounds and completion of required number of batches by the demand deadline. Figure 6.1 shows the plot for each of these values.

Graphs 6.1a and 6.1b both illustrate the results for the synchronised turnaround method, showing that the spread of OEE and CSL is quite vast but with the majority of campaign sequences lying below the 50% level for both metrics. The difference in

cost between the two shift patterns is however apparent with the cost associated with turnarounds dropping quite dramatically across the spread of campaign sequences. These trends are rather unsurprising due to the nature of the turnaround procedure; when in synchronised mode, the process is forced into a longer delay between campaigns because each room must wait for all rooms to finish processing the final batch before beginning turnaround activities. This means that the start of every new campaign is delayed thus affecting the ability to meet demand. The resulting 'penalty' is a decrease in the CSL value and since the performance part of OEE is based on the average CSL, there is a decrease in both metrics. Furthermore, when the shift pattern is 18hr7day the capacity of the facility diminishes relative to the 24hr7day pattern simply because the process is operational for fewer hours per day.
As the results in appendix C.1 and C.2 show the number of turnarounds are fewer because the number of campaigns processed reduces. Therefore the cost of changeover will show a decrease. This however is a trade-off; the cost of turnaround may decrease but this is because fewer campaigns have been processed thus showing a lower than desired OEE and CSL.

Graphs 6.1c and 6.1d show the results for the rolling turnaround procedure and immediately it is seen that there is a significant shift in CSL and OEE values to above the 50% mark. Again, this trend is as expected due to the fact that the delay between campaigns is decreased as the rooms are turned around as soon as possible. This minimises the time between the end of the last campaign and the start of the next, meaning that there is greater possibility of meeting demand deadlines, thus increasing the OEE and CSL values. For rolling turnaround, due to increased facility capacity there is less difference between the two shift patterns. Finally, as the number of campaigns achieved during the time horizon does not show variability between the two shift patterns (see Appendix C.3 and C.4) in rolling mode it is unsurprising that the associated costs do not show great variance between the two shift patterns, again attributed to the already increased capacity due to the decreased delay between campaigns.

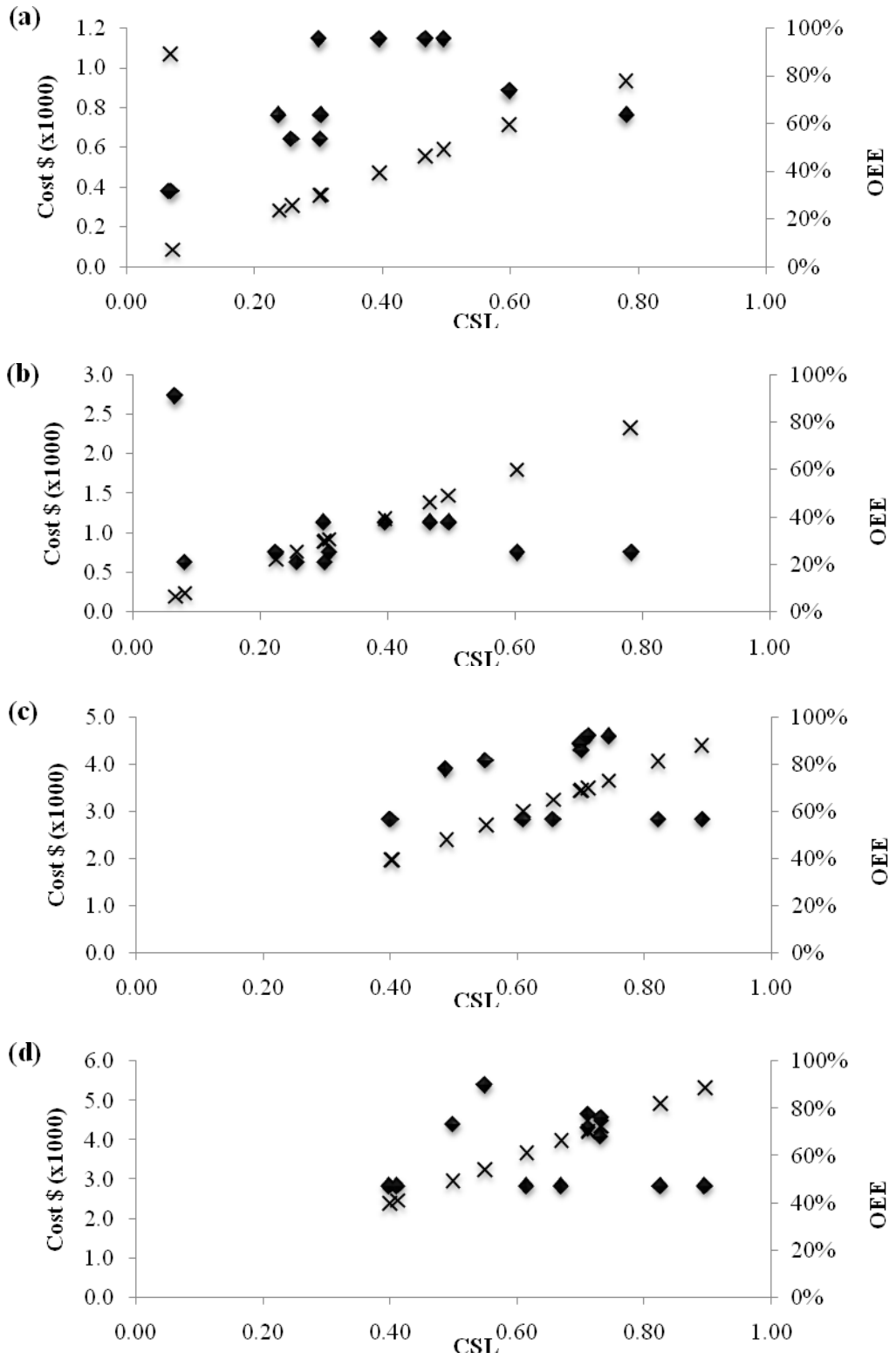


Figure 6.1 Cost of changeover, OEE and CSL for (a) Synch 18hr (b) Synch 24hr (c) Rolling 18hr (d) Rolling 24hr where ◆= Cost and x = OEE. Assuming no storage limit, zero failure rate and sufficient operators available to turn around all rooms in synchronised mode.

Extracting those campaign sequences from each of the four proposed strategies which show the highest CSL and OEE and the lowest cost (amongst the highest former values), it can be seen that the optimal sequence in each case is B-C-A. Figure 6.2 shows the metrics for this sequence under each turnaround strategy and shift pattern. Note that the metrics have been normalised in order to obtain comparable y-axis values. Furthermore, since the CSL and OEE values overlap the OEE values have been reduced by 10points so that they show on the graph. This does not affect the results as the graph is used for relative trends.

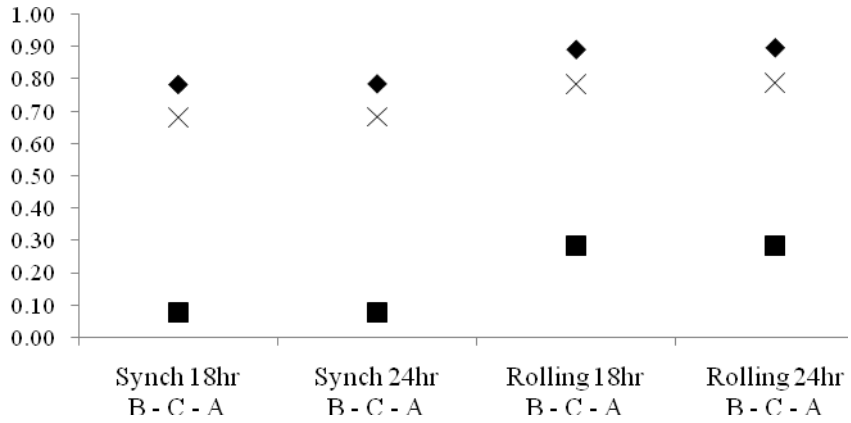


Figure 6.2 CSL, OEE and Cost of Turnaround for the optimal campaign sequences for each proposed strategy where (♦) = CSL, (■) = Cost, × = OEE. Assuming no storage limit for product, zero failure rate and sufficient operators available to turn around all rooms in synchronised mode.

As expected, the graph shows that the higher OEE and CSL values are attributed to the rolling turnaround mode. The associated costs are also higher because the number of campaigns achievable during the time horizon is higher. As the cost metric here is based solely on the cost of product changeover the two strategies show the same cost. However if the cost of labour were to be taken into account, the 24hr shift pattern would show greater cost, thus indicating another trade-off between process efficiency and cost.

6.6 Conclusion

This chapter has illustrated the use of the mAb model to act as a decisional tool in multiproduct process optimisation focussing on different turnaround strategies and operating shifts. Using Overall Equipment Effectiveness (OEE) and Customer Service Level (CSL) as the objective functions, it was found that given the demand profile used here, it would be more viable to process shorter campaigns first in order to achieve higher CSL values. What must be noted is that although demand profiles will be different across different products and across years, what this study has illustrated is the capability of the constructed tool in multi-product scheduling and analysis. Thus it has demonstrated the ability of the standard framework in guiding the construction of a model which meets the requirement specifications of relevance, reusability and sustainability.

CHAPTER 7

Commercial Considerations for the Development and Application of the Dynamic Standard Framework

7.1 Introduction

In this chapter, the potential commercialisation process is presented for the standard framework proposed in this thesis, focusing on its application in the area of biopharmaceutical manufacturing capacity management. Furthermore, it must be noted that while the market for commercialisation is mainly model developers, the end users of the models must also be taken into account when considering its commercialisation.

7.2 Application of the Standard Framework

The framework developed is primarily a methodology aimed to be used by modellers when approaching a modelling problem. The methodology consists of four major phases, Problem Structuring I, Problem Structuring II, Design and Construct, each of which guide the modeller in efficiently mapping the real system onto the chosen modelling platform in such a way as to meet the requirement specifications of a 'good' model. The methodology in its very nature is not platform specific and can guide any modelling practice across all industries. Thus the scope for application is vast.

However, the methodology has incorporated into it a set of templates which are designed to help in the construction phase of the modelling process and it is these templates which are platform specific, requiring a certain set of computer system requirements for their application. Figure 7.1 shows the infrastructure of the relationships within the application of the framework. Furthermore, these templates have been designed to be used for a specific scope, which the following section will discuss.

7.2.1 Application Area of Templates

Due to the extensive work carried out in building biotechnology based models, the templates have inevitably been developed for this application area. The parameters captured in the database, the output data generated and the template blocks themselves have been designed to deal with the complexities inherent to biopharmaceutical manufacturing capacity management. For example, production scheduling, single/multi-product procedures, ancillary activities and activities with associated cycle times.

7.2.2 Requirements for Template Application

The work carried out during the course of this research was specific to the Extend modelling environment, chosen due to the modelling preferences of the collaborative company, Eli Lilly and Corporation. The requirements for the use of these templates are as follows:

- The templates were created in Extend Version 7.06 and therefore must be used in version equal to or more recent than this. The lack of backward compatibility with this platform means that while the templates can be used in future versions, they cannot be used in older ones.
- Microsoft Excel must be available in order to use the database template. This input file as well as the output data file have been designed to automatically communicated with the Extend Platform and their presence is integral to the correct running of the model and block templates.

Similarly, if the model and its data capturing components are built in the Extend and Excel platforms, it is necessary for the end user to also hold a license for these platforms, following the specifications given above.

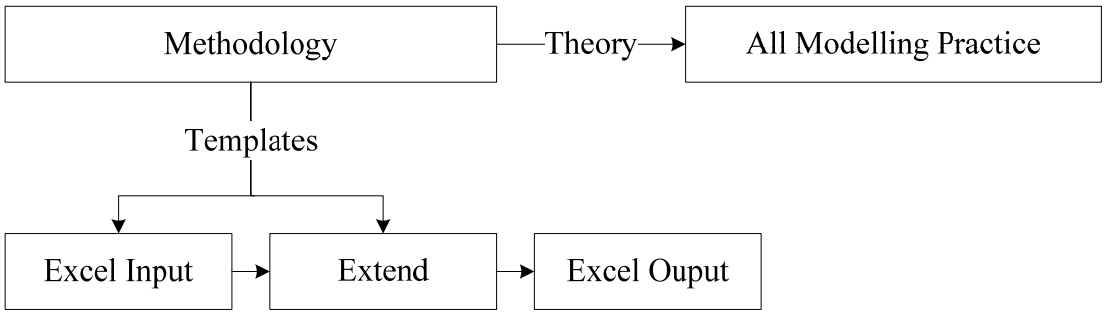


Figure 7.1 Infrastructure of Framework Application

7.3 Project Implementation

An example is given of an implementation scenario which involves a modeller (e.g. within a large-scale biopharmaceutical manufacturing company) being approached by the client (e.g. team running a new in house facility for the production of a biosynthetic therapeutic).

A chronological implementation plan is presented here of the key project phases as outlined in the proposed methodology.

7.3.1 Problem Structuring Phase I

- Client approaches modeller with capacity management issue
- Modeller establishes the scope of the problem, gaining an understanding of the desired inputs and outputs, the data available and the desired metrics to measure against i.e. what is the client aiming to measure process performance against. The list given in Table 3.4 (in Chapter 3) can be used to aid this.
- The model is then characterised based on the established requirements. For example, whether single or multiple activities, dynamic or static, metric etc. In this case, since the problem is associated with biopharmaceutical manufacturing

capacity management, it is assumed that the model can be classified as 'Extend Manufacture'.

7.3.2 Problem Structuring Phase II

- Modeller builds a non-coded description of the system using process terminology
- The system description is validated with client to ensure fully captured elements
- Description is tweaked and revalidated if necessary

7.3.3 Design

- Modeller creates a process specific description of the system, using model terminology. For example, tasks become sub-activities.
- Gaps highlighted here can be filled by returning to client. For example, constraints or cycle times which have not been captured.

7.3.4 Construct

- Modeller creates a coded description of the system, using the templates to map the system elements onto the modelling platform based on the model design.
- Debugging follows first phase of construct completion
- Using initial model outputs, validation can take place, comparing outputs with expected results from client

7.3.5 Project Handover

- Model is presented to client followed by training on how to use the model
- Project is handed over to client along with documentation

7.4 Project Costing

The resource requirements and costs have been estimated based on a typical industrial project. The assumptions made are as follows:

- Although both modeller and client are in-house, the modeller is assumed to be working on the project as a consultant, using an opportunity cost basis for determining costs associated with time spent on this project. The costs for the modeller have been assumed to be £20 per hour (lower than general consultancy rates due to in-house employment status), working a 40 hour week.
- It is assumed that the modeller holds a license for the required modelling platforms. Therefore the client will incur the cost of a run-time license fee, which will not include maintenance or support fees.
- Both the modeller and client own licenses for Microsoft Excel and therefore no costs are incurred
- Table 7.1 shows the estimated project cost to total £8,080 with the project duration being approximately 50 days.

Table 7.1 Project costing and task durations

Task	Duration	Resource	Cost (£GBP)
Problem Structuring Phase I	2 days	Modeller	320
Problem Structuring Phase II	10 days	Modeller	1,600
Design	5 days	Modeller	800
		Microsoft Excel	-
Construct inc debug	30 days	Modeller	4,800
		Extend	-
		Microsoft Excel	-
Project Handover	3 days	Modeller	480
		Extend	80
		Microsoft Excel	-
Total	**50 days**		**8,080**

7.5 Potential Benefit of Framework Application

The benefits of the standard framework are two-fold. Firstly, the increased efficiency in model construct means that the modeller can move on to the next project in a far quicker time frame, thus increasing their productivity. More importantly the client will receive the answers to their capacity management problems far more efficiently, thus allowing for quicker implementation of any recommended changes to the process. Mallik et al (2002) estimate that for a typical new mammalian cell culture facility, even a 25% increase in capacity utilisation can result in an average of $280 million increase in its net present value. Thus, the possible savings through the use of the standard framework are potentially very significant, highly beneficial to an industry somewhat struggling to maintain high profit margins.

7.6 Dissemination of Standard Framework

The potential benefit of the standard framework can only be realised if it is disseminated across the biopharmaceutical industry as a support tool. As with all standards, its adoption relies on widespread knowledge and user reviews. In industries such as the biopharmaceutical industry, the best means of reaching the target audience is through conference presentations, publications and release into the modelling community. Upon completion, the framework presented in this thesis was handed over to the sponsor company to be used in their current and future modelling projects. The positive feedback received acted as a key milestone indicating that dissemination to industry users had been achieved in the project.

CHAPTER 8

Validation Issues

8.1 Introduction

The EMEA (The European Agency for the Evaluation of Medicinal Products) defines process validation as “the act of demonstrating and documenting that a procedure operates effectively....the means of ensuring and providing documentary evidence that processes (within their specified design parameters) are capable of consistently producing a finished product of the required quality”. Similarly, the FDA (Food and Drug Administration) defines validation as “Establishing documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its pre-determined specifications and quality attributes”. Both governing authorities therefore require that a process consistently meets a certain degree of quality. In order to ensure this, there are guidelines in place in the form of GMP or “Good manufacturing practice" which refers to the quality control of manufacturing for foods, pharmaceutical products, and medical devices. These guidelines are part of pharmaceutical legislature in many countries although the details of each guideline may differ from country to country. Process validation guidelines such as GMP primarily deal with critical aspects of a process, that is, those aspects which directly affect the quality of product.

The work in this thesis outlines the process of model construct which can be considered an off-line process which can only affect the manufacturing process indirectly if changes are made as a result of the simulations. In such cases, the client must ensure that any process changes directly affecting the quality of the product are correctly tested and subsequently validated. Consistency lots will be required to test not only the product quality but also the overall efficiency of process; there are documentations which must verify that all aspects of the installation meet the

manufacturer's specifications, that when assembled and in use all equipment within the process do perform as defined by the manufacturers specifications and that the performance of the equipment meet requirements. These documents are the Installation (IQ), Operation (OQ) and Performance Qualifications (PQ) and are vital in achieving equipment/process validation.

Due to the nature of the work presented in this thesis, software validation must be considered and is the focus of the following section.

8.2 Software Validation

The standard framework presented in this thesis is based on the use of existing software. According to GMP guidelines as defined under the FDA, "Validation requirements apply to software used as components in medical devices, to software that is itself a medical device, and to software used in production of the device or in implementation of the device manufacturer's quality system". Thus the framework does not fall under this validation category.

8.3 Decision Support Systems

Decision support systems are a class of information systems which support decision making activities. As stated in the previous section, the work presented in this thesis does not fall under the conventional category of software validation, as it is not designed to be used as an on-line process tool. However validation is required in the form of software testing which decision support systems generally go through, albeit in different forms. In the case of this thesis, the subject of tool validation was briefly discussed in the previous chapter and is an integral part of the methodology proposed, stating that at each stage of the modelling process, understanding of the system elements, model design and final construct must be validated with the client. This dynamic approach to the modelling process will:

- Ensure that the initially determined scope is well understood and relevant

- Ensure that all relevant systems elements such as constraints, rules, and parameters are sufficiently captured
- Allow for debugging of the model

8.4 Conclusion

Due to the nature of the work carried out in this thesis, validation is not a legal requirement. However in order to ensure that the models created as a result of the application of the standard framework are useful in their capacity as decision support systems, it is necessary to follow the validation or testing protocols stipulated at each stage of the methodology.

CHAPTER 9

Conclusions and Future Work

9.1 Introduction

Capacity management in the manufacture of biopharmaceuticals has changed over the years, with a clear shift towards creating more cost effective processes (Farid 2009). As such some biopharmaceutical manufacturers are seeking to use simulation tools to experiment with various process alternatives at various stages of a facility's life, from inception and design to ongoing process development. However in order for simulation techniques to add value as decisional support tools it is necessary to standardise the way in which the modelling process is approached. The aim of this thesis was to propose a dynamic standard framework to facilitate the practice of simulation modelling in biopharmaceutical capacity management. A methodology was developed including a set of standard templates as part of this framework. Section 9.2 highlights the significance of the work while Section 9.3 summarises the efforts made in this thesis to develop the framework and the overall conclusions made. Section 9.4 suggests directions for future research work.

9.2 Significance of Work

The platforms available for dynamic discrete event simulation provide an extensive library of blocks and coding languages to aid the modelling process, allowing for highly complex representations of the systems which they model. However the myriad of ways in which these tools can be used results in a complexity across the modelling portfolio of a company which makes them unintuitive and unsustainable. For example, the mAb case study presented in Chapter 5 was based on two pre-existing models which were built during the design stages of the process. Although

these models were built by two colleagues sharing an office, they differed significantly in design, data input and output approaches, modelling techniques used and coding practices. Furthermore, upon reverse engineering these models it was discovered that they were modelling exactly the same system elements and answering almost identical questions. In other words they were performing the same functions with neither meeting the requirement specifications proposed in this work. If such differences can occur within an office, the problem is even greater across sites and geographical locations. Therefore a standard is required to streamline modelling activities. However, as found during an extensive literature review, the current standards for modelling practice are simply guidelines for the graphical representation of real systems and the annotations used. For example, how many hierarchical levels should be used, the connections between the system elements and their nomenclature. These guidelines aid the modelling process to a minimal degree, offering little help on the complexities such as how to capture rules, constraints, uncertainties and elements of biopharmaceutical systems such as batch cycling and buffer make-up scheduling. Therefore the non-existence of standards has increased the difficulties surrounding the simulation process, particularly in the biopharmaceutical industry which only recently adopted simulation techniques relative to other industries such as the semiconductor industry. Hence the significance of the work presented in this thesis is highlighted.

9.3 Overall Conclusions

The main focus of this thesis has been to develop a standard framework for the development of simulation models to effectively function as decisional tools. In order to understand the sorts of capacity management issues these decisional tools must tackle, Chapter 1 reviewed the challenges faced by biopharmaceutical manufacturers, in particular, process uncertainties, multi-product scheduling and capacity utilisation. A review of the different types of simulation environment was carried out in Chapter 2 along with a review of their application in the biopharmaceutical and other industries so that an understanding could be reached of the best environment or platform on which to focus the work in this thesis. It was concluded that discrete event simulators were the preferred platform for creating

dynamic models capable of representing uncertainties and constraints along with a more user friendly modelling environment. Furthermore the need for standardisation was recognised in order to aid the modelling process and as such the current standardisation approaches available were reviewed to assess whether an existing standard could be applied to the problem. It was concluded that only a few components of these standards could be applied as they did not take into account the many complexities of biopharmaceutical manufacture.

The standardisation approach outlined in this thesis provides the methodology and modelling tools to allow modellers to build models which meet the requirement specifications of 1) Intuitive to user, 2) Relevance, 3) Ease of Data Input/Output, 4) Short Run Time, 5) Maximised Reusability and Sustainability, and 6) Minimised Development Time, thus creating valuable and cost effective decisional tools. The main premise of the methodology is to aid during the data gathering and model design stages in order to improve the efficiency of data retrieval and validation. Standard templates developed as part of the framework provide the fundamental building blocks, with a degree of customisation which will allow different process elements to be realistically and more easily captured, but with sufficient generality built in to allow them to be used across different system models. The domain focussed on in this work is the 'Process' level of the organisational hierarchy, more specifically biopharmaceutical batch processes, looking at the manufacture of bulk products, with the main system elements captured being resources (e.g. labour, buffer, equipment), entities (e.g. batches) and activities (e.g., product handling, CIP). The features of biopharmaceutical manufacture captured in the work will be discussed below and include upstream fermentation, downstream recovery and purification at the unit operation level, mass balancing based on step yield and equipment sizing based on upstream titre. Also resource scheduling such as 'just-in-time' buffer makeup, equipment failure, batch splitting for purification and multiproduct scheduling. The platform chosen for the work is Extend (ImagineThat!, CA), a discrete event simulation software with a graphical user interface.

The development of this standard framework was illustrated in Chapter 3 as part of an evolutionary process with reference to case studies carried out to test the various versions of the standard during its development.

The first of these case studies was presented in Chapter 4 where the first version of the standard was applied to the construction of a model for a biopharmaceutical manufacturing system, referred to as BioSynT, testing the ability of the standard to guide the construction of a manufacturing capacity management model capable of meeting the requirement specifications. The scope of the study was to determine how fast the downstream production process for a biosynthetic therapeutic could go, for a fixed batch demand, given certain constraints such as operating rules and uncertainties such as equipment failure. Firstly a deterministic sensitivity analysis was carried out to identify the key parameters that affect the overall process cycle time. Using these parameters Monte Carlo simulations were run to perform debottlenecking analyses and to understand the impact of different scenarios such as additional labour or process equipment, combined with process uncertainties to determine how these would affect the running of the process, and to highlight any shifts in bottlenecks. A comparison of scenario output distributions allowed for this. A cost analysis of the scenarios was also carried out to add a further dimension to the study and to understand the trade-offs between operational improvements and the cost of process changes. Secondly a qualitative analysis showed that although the model was able to help perform useful analyses, the requirement specifications were not sufficiently met, with rigidity in the method of model construction and the approach used leading to a less than adequate reusability and sustainability, a relatively long run time and a far greater development time than desired (with time and cost efficiency as measures).

Chapter 5 presented the second case study, that of a single product monoclonal antibody (mAb) manufacturing system, including both upstream and downstream processes, to test the second version of the standard framework. A similar study to that applied to the BioSynT case was carried out, combining deterministic, stochastic (Monte Carlo) and cost analyses to determine the impact of process uncertainties and different scenarios (such as new chromatography columns or resin upgrades) on the chosen output parameter, process throughput. To add another dimension to the application of the model and to take full advantage of the system complexities built into it, an addition to the mAb case study was introduced as presented in Chapter 6, looking at schedule optimisation for a multiproduct scenario focussing on different

turnaround strategies and operating shifts. Using Overall Equipment Effectiveness (OEE) and Customer Service Level (CSL) as the objective functions the model was used to run different scenario combinations to determine the best scheduling strategy. The studies presented in Chapters 5 and 6 demonstrated the ability of the standard framework in guiding the construction of a model which could be used for complex analyses, meeting the requirement specifications of relevance, reusability and sustainability. Furthermore, a qualitative analysis showed that these were met to a far greater degree than illustrated with the BioSynT model, with significant improvements in model development time, run-time, user intuitiveness, and ease of data input/output. However the need for a more comprehensive library of standard building blocks was also identified in order to help further accelerate the time spent on model development, to facilitate debugging through a more standardised block structure and to improve model intuitiveness. Thus the final version of the standard was developed.

The work in this thesis highlights the benefits of adopting a standardised approach to simulation modelling in biopharmaceutical manufacturing capacity management. This finding has been illustrated through a series of capacity management case studies. The methodology developed can be used as a framework to guide the modelling process, allowing for a greater understanding of the system being modelled, a faster data validation process, and ultimately the construction of a model which meets the six requirement specifications. In doing so, the use of the standard framework tool can save a company time and money, adding value as a decisional tool at various stages from process design to full operation.

9.4 Future Work

The standard framework and tools presented in this thesis contribute to the growing field of simulation modelling in biopharmaceutical manufacture. It also provides a basis for future work as discussed next.

The domain focussed on in this work has been the ‘Process’ domain, specifically looking at bulk manufacture. However as described in Chapter 3 there are a further six domains – Supply chain, Enterprise, Product, Site, Dept/Building and Equipment,

which could be considered. Although the methodology itself is a universal one, the templates have been specifically created for biopharmaceutical manufacturing capacity management models. An in-depth study of the commonalities and complexities across the different domains could lead to the expansion of the templates thus allowing for the framework to be used across an entire organisation for all systems. Furthermore, similar studies at the product and process level domains could allow for expansion into different areas of manufacturing, for example QC labs or Fill/Finish activities.

Many different capacity management questions were considered in this work, ranging from equipment utilisation to process changes and operational rules. However, further case studies could be used to test the ability of the standard to help create decisional tools capable of answering more questions such as the impact of using disposables. To test the reusability and sustainability of the existing models, the scope of those case studies could also be expanded. Furthermore, the models built were created with some modelling constraints such as the ability of the mAb model to handle only three different products at a time in a multiple product scenario. The models could therefore be expanded on to be more versatile. For example, the database could be constructed to hold data for 10 different products, with all parameters such as specific step yield being highly variable according to the product mix.

The reviews carried in this work have been very qualitative in nature. A more quantitative analysis to measure the exact value added could be carried out by embarking on a case study whereby the same system model would be built in two instances, one where the methodology and templates were implemented and one where they were not. This would give a direct comparison of the times attributed to design, development and validation, and the corresponding cost of developer hours and client consultations.

The economic analyses carried out have been very basic, taking into account only the base costs of process alternatives such as membrane upgrades or labour increase. A more detailed process economics calculation would give a far greater understanding of the cost implications of the proposals and also increase the usefulness of the

models built. A way of doing this could be to attach the models to a cost database, either integrated into the main database template or in addition to it, providing a cost profile of all inputs to the model. Outputs such as process throughput could also be economically evaluated post simulation in order to provide an understanding of return on investment.

The software used in this work, ExtendSim v7, was found to be appropriate for modelling biopharmaceutical manufacturing models. Although the templates were built specific to this platform, the methodology provides a universal approach and therefore its application to other modelling platforms and environments such as different discrete event simulators or optimisation tools could test its usefulness as a modelling standard.

The modelling methods considered use a non-database driven approach whereby the database only provides the system parameters. This approach is aimed more at the developer as the model itself requires design and construction, albeit with the aid of the standard framework. Conversely, database driven modelling can be described as the modelling of a system whereby data input to the database determines the physical and logical structure of the model along with the system parameters. This approach is aimed more at the end user and the modelling constructs already exist but do not have identity or structure until the database is populated. Unlike non-database driven modelling where built-in commonality exists but any higher levels of detail can be supplied by the developer to tailor the model to the specific system, the level of standardisation required is far greater in the database driven approach as the constructs must be able to take the identity of any system element and every possible scope or question must be built in beforehand. However the latter approach does minimise interaction with the model and can potentially reduce development time significantly, eventually negating the need for a model developer at all once the constructs have been fully put into place. Studies of all biopharmaceutical domains, their possible scopes and an understanding of their commonalities can help move the standard framework towards a more database driven methodology.

In conclusion, the future work outlined draws upon the framework and methods established in this thesis. The development of more sophisticated models and

expansion across different domains would enhance the standard framework's capacity as a guide to create valuable decisional tools. The lack of standardisation methods in the biopharmaceutical industry means that tools such as the framework described in this thesis are much needed and their use will be widely adopted across the industry in the future.

REFERENCES

Aldridge, S., (2009).Bringing greater efficiency to antibody manufacturing, *Genetic Engineering and Biotechnology News*.**29**, 14

Al-Ahmari, A.M.A. and Ridgway, K., (1999). An integrated modelling method to support manufacturing systems analysis and design, Computers in Industry, 225–238

Banks, J., (1998). Handbook of simulation, Engineering and Management Press, John Wiley Sons Inc

Banks,J et al., (1991). The Simulator: New member of the simulation family, *Interfaces*. **21**, 76-86

Barjis, J. and Shishkov,B.,(2001). UML based business systems modelling and simulation, in: Proceedings of the Fourth International Eurosim 2001 Congress, Delft, The Netherlands

Benson,D.,(1997). Simulation modelling and optimization using ProModel, *Proceedings of the Winter Simulation Conference,* edsS. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 587-593, Orem UT

Berg, E.L, et al., (2005). Biological complexity and drug discovery: a practical systems biology approach, IEE Proc.-Syst. Biol. **152**, Dec 2005

Biwer,A., Griffith, S., Cooney,C., (2005). Uncertainty analysis of penicillin V production using Monte Carlo simulation.*BiotechnolBioeng*. **90**, 167-179

Bosch,P.C., (2003).HighMAST™ Versus "Workbench" Style Simulation Tools, Highpoint Software Systems, Waukesha WI

Brooks,R.J. and Robinson,S., (2001). Simulation, with Inventory Control (author C. Lewis), Operational Research Series, Palgrave: Basingstoke.

Collier,R., (2009).Drug development cost estimates hard to swallow, Canadian Medical Association, Feb 3, 180

Connolly, A., (2001). Biotech companies face capacity crunch

DiMasi, J.A et al., (1991). Cost of innovation in the pharmaceutical industry. *Journal of Health Economics.***10**, 107–142.

DiMasi, J.A et al.,(2003). The price of innovation: new estimates of drug development costs, *Journal of Health Economics***22**, 151–185

Farid.S, et al., (2000). A hierarchical framework for modelling biopharmaceutical manufacture to address process and business needs, *Computer Aided Chemical Engineering*, **8**, 673-678

Farid.S, et al., (2006). Modelling biopharmaceutical manufacture: Design and Implementation of SimBiopharma, *Computers and Chemical Engineering*

Farid, S., (2009), Process Economic Drivers in Industrial Monoclonal Antibody Manufacture, Process Scale Purification of Antibodies, Edited by Uwe Gottschalk. 239-261

Federal Information Processing Standards Publication 183, Integration Definition for Function Modelling (IDEF0), Dec 1993

Fonendo, www.fonendo.com/noticias/41/2002/09/1

Fox, S. et al., (2001). A Time to Build Capacity

Geuder,D.,(1995). Object-oriented modelling with Simple++, *Proceedings of the 1995 Winter Simulation Conference*, edsC.Alexopoulos, K.Kang, W.R.Lilegdon, and D.Goldsman, Association for Computing Machinery, New York.534-540.

Gibson, R., Dickson, C., Simulation Software Practical Guidelines for Approaching the Selection Process, Automation Associates Inc

Glinz, M., (2000). Problems and deficiencies of UML as a requirements specification language, InstitutfürInformatik, Universität Zürich

Gosling, I., (2003). Optimising Productivity in Multiproduct Batch Facilities, Genetic Engineering News

ISA 1995, Batch Control: Part 1 Models and Terminology

ISA-95.00.01- 2000, Enterprise Control Integration: Part 1 Models and Terminology

Hauge.J.,Paige.K., (2001) Learning SIMUL8: The Complete Guide, PlainVu Publishers

Johnston, R.,(2010). Simulation tools in the optimisation of biopharma manufacturing processes, Biomanufacturing, Innovations in Pharmaceutical Technology

Joines, J.A., Roberts,S.D., (1998). Handbook of simulation, Object Oriented Simulation,, Engineering and Management Press, John Wiley Sons Inc. 397-428

Jully et al., (2004).Modelling, optimization, and debottlenecking of a pharmaceutical production process utilizing a batch process simulator, The Malaysian Society for Molecular Biology and Biotechnology Scientific Meeting

Kamarck, M.E., (2006). Building biomanufacturing capacity - the chapter and verse, *Nature Biotechnology***24**, 503-505

Kelasky,D.R. and Levassuer,G., (1997). Using SIMPLE++ for improved modelling efficiencies and extending model life cycles, *Proceedings of the 1997 Winter Simulation Conference*, edsP. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 701-708, Madison WI

Knutilla, A, et. al., (1998). "*Process Specification Language: Analysis of Existing Representations*," NISTIR 6133, National Institute of Standards and Technology, Gaithersburg, MD

Krahl,D., (2000). The Extend simulation environment, *Proceedings of the Winter Simulation Conference*, edsJ.A.Joines, R.R.Barton, K.Kang, and P.A.Fishwick, 280-289, San Jose CA

Lakhdar.K. et al., (2005).Medium term planning of biopharmaceutical manufacture using mathematical programming,*Biotechnol. Prog.***21**, 1478-1489

Langer, E.S., (2004). Big shifts in outsourcing of biopharmaceutical manufacturing, *Genetic Engineering News*

Langer, E.S., (2008).Batch Failure Rates in Biomanufacturing, *Genetic Engineering & Biotechnology News*.**28**

Law, A.M. andKelton, D.W., (1991).Simulation Modeling& Analysis, McGraw-Hill International Editions.1-20

Levasseur, G.A., (1996).The Case for object-oriented simulation modelling, OR/MS Today, Aug 1996.65-67

Li et al., (2005). A Framework for Standard Modular Simulation: Application to Semiconductor Wafer Fabrication Systems, *Proceedings of the Winter Simulation Conference*, eds M.E. Kuhl, N.M.Steiger, F.B. Armstrong and J.A. Joines, 2162 – 2171

Lim, A.C, et al., (2004).A decisional-support tool to model the impact of regulatory compliance activities in the biomanufacturing industry, *Computers and Chemical Engineering*.**28**, 727–735

Myers,K.L. and Wilkins,D.E., (1997). The Act Formulism. Version 2.2, SRI International Artificial Intelligence Center, Menlo Park, CA, Sept 1997

Mallik, A, et al., (2002).Biopharma's capacity crunch

McLean,C. and Leong, S.,(2002). A framework for standard modular simulation, *Proceedings of the Winter Simulation Conference*, eds E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1614 – 1620, Gaithersburg, MD

McKenzie, P et al., (2006). Can pharmaceutical process development become high tech?,*AlChe Journal*. **52**, 3990 – 3994

Miller, V., (2002). Things Go Better with S-88: A new approach to the engineering process, *Pharmaceutical Engineering*. **2**

Miller,D.L et al., (2010). Mathematical programming for the design and analysis of a biologics facility-Part 2.*Biopharm Int*. **23**

OMG.,(1999).*OMG Unified Modeling Language Specification Version 1.3.* OMG document ad/99-06-08.ftp://ftp.omg.org/pub/docs/ad/99-06-08.pdf

Onggo, B.S.S., (2009). Towards a unified conceptual model representation: a case study in healthcare, *Journal of Simulation*. **3**, 40–49

Oscarsson, J. and Moris, M.U., (2002). Documentation of discrete event simulation models for manufacturing system life cycle Simulation, *Proceedings of the Winter Simulation Conference*, edsE. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1073 – 1078, Skovde, Sweden

Pandeo, www.pandeo.com

Pandiana,Y et al.,(2002).A well structured method for modelling batch-wise industrial processes according to the S88 standard.TechnischeUniversiteit Delft,

Petrides, D.P et al., (2002).The role of process simulation in pharmaceutical process development and product commercialisation, *Pharmaceutical Engineering.***22**

Petrides, D.P. andSiletti, C.A.,(2004). The role of process simulation and scheduling tools in the development and manufacturing of biopharmaceuticals, *Proceeding of the Winter Simulation Conference*, eds R.G. Ingalls, M.D. Rossetti, J.S. Smith and B.A. Peters, 2046-2051, NJ

Pharmacos, Manufacture of biological medicinal products for human use. www.pharmacos.eudra.org/F2/eudralex/vol-4/pdfs-en/anx02en.pdf

Polastro, E.T.,(1996). Managing primary process development. Pharmaceutical Manufacturing International, P.A. Barnacal, Sterling Publications Limited, London, 67-70

Rajapakse, A., Tictchener-Hooker, N,J. andFarid,S., (2005). Modelling of the biopharmaceutical drug development pathway and portfolio management.*Computers and Chemical Engineering*.**29**, 1357–1368

Ramos, A.L. and Vilarinho, P.M., (2003). Modelling a simulation tool for cellular manufacturing systems, IEPM'03 -International Conference on Industrial Engineering and Production Management, Porto Portugal

Ransohoff, T.C., (2004). Consideration impacting the of make v.s buy decision, American Pharmaceutical Outsourcing

Robinson, S., (2008). Conceptual modelling for simulation Part 1: definition and requirements, *Journal of the Operational Research Society*. **59**, 278-290

Roth, G.Y., (2001). Biopharmaceutical Outsourcing – Will a supply crunch hurt this growing market? www.contractpharma.com/jun012.htm

Ryan,J. and Heavey, C., (2006). Process modeling for simulation, *Computers in Industry*.**57**,437-450

Sadowski, D.A. and Grabau, M.R. (2000). Tips for successful practice of simulation, *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines et al., IEEE: New York, pp. 26-31.

Sinnott, R.K., Coulson, J.M. andRichardson, J.F., (2005).Coulson and Richardson's Chemical Engineering: Chemical Engineering Design,Butterworth-Heinemann. **6**,265

Schlenoff,C et al., (2000). The Process Specification Language (PSL): Overview and Version 1.0 Specification

Seila, A.F., (2001). Spreadsheet simulation, *Proceeding of the Winter Simulation Conference*, edsB. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 74-79, Athens, GA

Seila, A.F., (2005) The Case for a Standard Model Description for Process Simulation, International Journal of Simulation and Process Modelling. **1**, 26-34

Siau, K. and Cao, Q., (2001).Unified Modeling Language (UML) – A Complexity Analysis.J.Datab.Manage.**12**, 26-34

Sharda,B. and Bury, S.J, (2008). A discrete event simulation model for reliability modeling of a chemical plant, *Proceedings of the 2008 Winter Simulation Conference*, Freeport, TX

Takus, D.A. andProfazich,D.M., (1997). Arena Software Tutorial, *Proceedings of the Winter Simulation Conference*, edsAndradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 541-544, Sewickley, Pennsylvania

Taylor˙ www.taylor.com/industry/pharmaceutical

Uschold, M. and Gruninger M., (1996).Ontologies: Principles, Methods, and Applications, *Knowledge Engineering Review*. **11**, 96-137.

Valentin, E.C. and Verbraeck, A., (2002). Guidelines for designing simulation building blocks**,** *Proceedings of the 2002 Winter Simulation Conference* (ed. *E. Yücesan,et al), 563-571*

Valentin, E.C. andVerbraeck, A., (2002). Simulation using building blocks. In: F. Barros ; N. Giambiasi (Eds.) *Proceedings conference on AI*, Simulation and Planning, San, Diego: SCS. 65-70.

Walker,R. Using ISA95 to Automate Manufacturing Operations, Dupont Engineering

Wang,W. and Brooks, R.J., (2007). Conceptual modelling and the modelling process in simulation projects: A Survey of Simulation Modellers,

Winder.R., (2005). Cell culture grows capacity, Chemistry and Industry

(www.mel.nist.gov/psl) PSL website

http://www.idef.com/idef0.html

http://en.wikipedia.org/wiki/Unified_Modeling_Language

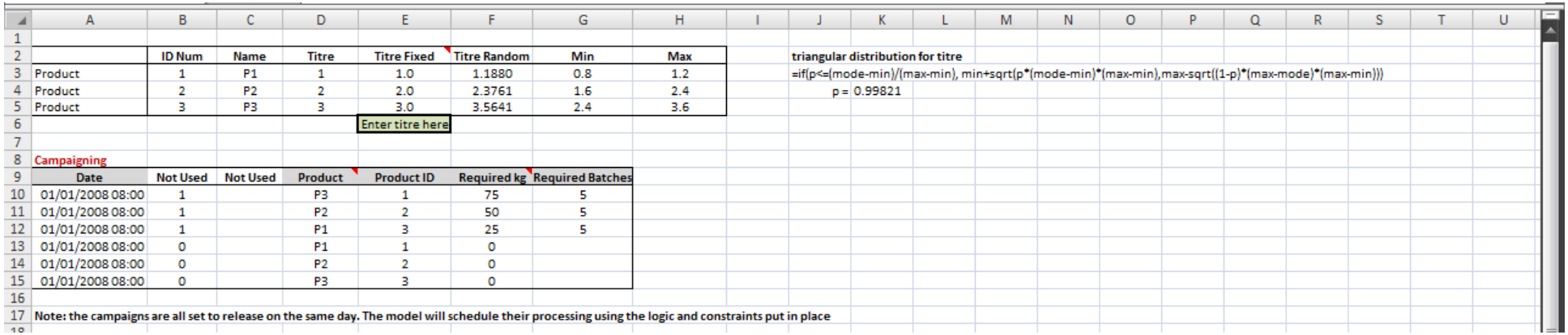
http://www.batchcontrol.com/s88/01_tutorial/04-recipes.shtml

APPENDIX A

Data for Chapter 3

Appendix A.1 Standard Framework Version 3: Data Input Template

Activities Tab



	ID Num	Name	Titre	Titre Fixed	Titre Random	Min	Max
Product	1	P1	1	1.0	1.1880	0.8	1.2
Product	2	P2	2	2.0	2.3761	1.6	2.4
Product	3	P3	3	3.0	3.5641	2.4	3.6

Enter titre here

triangular distribution for titre

=if(p<=(mode-min)/(max-min), min+sqrt(p*(mode-min)*(max-min),max-sqrt((1-p)*(max-mode)*(max-min)))

p = 0.99821

Campaigning

Date	Not Used	Not Used	Product	Product ID	Required kg	Required Batches
01/01/2008 08:00	1		P3	1	75	5
01/01/2008 08:00	1		P2	2	50	5
01/01/2008 08:00	1		P1	3	25	5
01/01/2008 08:00	0		P1	1	0	
01/01/2008 08:00	0		P2	2	0	
01/01/2008 08:00	0		P3	3	0	

Note: the campaigns are all set to release on the same day. The model will schedule their processing using the logic and constraints put in place

Scheduling Tab

Operating Shifts

Form shifts by inputting: hrs/day, days/week and start time (max two shifts)			
Shift	**Hours per day**	**Days per week**	Start Time
Shift12hr7d	12	7	06:00:00
Shift12hr7d	12	7	12:00:00

Assumption: only two shift will be necessary for a full operating day
Any overlaps modelled indirectly through the start time inputs
The number of operators available input under General Parameters tab

Operating Shifts (automatically filled in, no user input necessary)

Shift 1			Shift 2		
Time	**Operators**	**On/Off (1/0)**	**Time**	**Operators**	**On/Off (1/0)**
0	3	1	6	3	1
12		0	18		0
24	3	1	30	3	1
36		0	42		0
48	3	1	54	3	1
60		0	66		0
72	3	1	78	3	1
84		0	90		0
96	3	1	102	3	1
108		0	114		0
120	3	1	126	3	1
132		0	138		0
144	3	1	138	3	1
144		0	138		0
144	3	1	138	3	1
144		0	138		0
144	3	1	138	3	1
144		0	138		0
144		1	138		0

MAIN Facility Shutdown (resulting in no utilities)

Date	ON/OFF	Duration (days)
01/01/2008 00:00	1	
07/04/2008 00:00	0	0
07/04/2008 00:00	1	
26/08/2008 00:00	0	14
09/09/2008 00:00	1	
19/12/2008 00:00	0	7
26/12/2008 00:00	1	
29/12/2008 00:00	0	0
29/12/2008 00:00	1	

Enter the duration and start date of the scheduled shutdown of the main facility. The end date (purple) will automatically fill
If shutdown is not needed, simply zero the duration

RUN MODEL | General Parameters | Activities | Products | **Scheduling** | Process Areas | Equip Information

Process Areas Tab

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2				Shutdown					Turnaround					
3	**Area**	**Description**	**No. of equip in room**	**Scheduled Shutdown (days)**	**Shutdown Time (hrs)**		**Synchronised SD**	**SD Setup Check**	**Activity 1**	**Activity 2**	**Activity 3**	**Activity 4**	**Activity 5**	**Activity 6**
4	Room1													
5	Room2													
6	Room3													
7	Room4													
8	Room5													
9	Room6													
10	Room7													
11	Room8													
12	Room9													
13	Room10													
14	Room 11													
15	Room 12													
16														
17	**Shutdown = scheduled shutdown of system.**													
18	**Turnaround = activities associated with switch between products in a multiproduct system.**													
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														
32														
33														
34														

RUN MODEL | General Parameters | Activities | Products | Scheduling | **Process Areas** | Equip Information

Equip Information Tab

	A	B	C	D	E	F	G	H	I	J
2	**GENERAL**					Yield			Stability	
3	**Equip/Activity**	**Room**	**Section**	**Volume**	**Yield(P1)**	**Yield(P2)**	**Yield(P3)**	**Stability(P1)**	**Stability(P2)**	**Stability(P3)**
4	Activity 1		1		1	1	1	1	1	1
5	Activity 2		1		1	1	1	1	1	1
6	Activity 3		1		1	1	1	1	1	1
7	Activity 4		1		1	1	1	1	1	1
8	Activity 5		1		1	1	1	1	1	1
9	Activity 6		1		1	1	1	1	1	1
10	Activity 7		1		1	1	1	1	1	1
11	Activity 8		1		1	1	1	1	1	1
12	Activity 9		1		1	1	1	1	1	1
13	Activity 10		1		1	1	1	1	1	1
14	Activity 11		2		1	1	1	1	1	1
15	Activity 12		2		1	1	1	1	1	1
16	Activity 13		2		1	1	1	1	1	1
17	Activity 14		2		1	1	1	1	1	1
18	Activity 15		2		1	1	1	1	1	1
19	Activity 16		2		1	1	1	1	1	1
20	Activity 17		2		1	1	1	1	1	1
21	Activity 18		2		1	1	1	1	1	1
22	Activity 19		3		1	1	1	1	1	1
23	Activity 20		3		1	1	1	1	1	1
24	Activity 21		3		1	1	1	1	1	1
25	Activity 22		3		1	1	1	1	1	1
26	Activity 23		3		1	1	1	1	1	1
27	Activity 24		3		1	1	1	1	1	1
28	Activity 25		3		1	1	1	1	1	1
29										
30										
31	**RELIABILITY**	P1				P2				
32	**Equip/Activity**	**P(Failure)**	**Fail Occurance**	**Time to Repair**	**Discard Rate**	**Equipment**	**P(Failure)**	**Fail Occurance**	**Time to Repair**	**Discard Rate**
33	Activity 1									
34	Activity 2									
35	Activity 3									
36	Activity 4									
37	Activity 5									
38	Activity 6									
39	Activity 7									
40	Activity 8									
41	Activity 9									

	L	M	N	O
		Overall Yield For Process Sections		
	Section	**P1**	**P2**	**P3**
	1	1.000	1.000	1.000
	2	1.000	1.000	1.000
	3	1.000	1.000	1.000
	4	0.000	0.000	0.000
	5	0.000	0.000	0.000
	rooms (linked to Process Areas tab for listing)			
			Room1	
			Room2	
			Room3	
			Room4	
			Room5	
			Room6	
			Room7	
			Room8	
			Room9	
			Room10	
			Room 11	
			Room 12	

RUN MODEL | General Parameters | Activities | Products | Scheduling | Process Areas | Equip Information

Column Info Tab

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2														
3										(Fixed cycles)				
4	Column Parameters	Data Fed to Extend								Fixed				
5	Column	Height (cm)	Diameter (cm)	Volume (L)	Cycles after repack	Protein	dbc (Product 1)	dbc (Product 2)	dbc (Product 3)	Height (cm)	Res Time (min)			
6	Column 1	20	60	57	0	Protein 1	30	20	50	25				
7	Column 2	20	60	57	0	Protein 2	30	20	50	20				
8	Column 3	20	60	57	0	Protein 3	30	20	50	30				
9	Column 4	20	60	57	0	Protein 4	30	20	50	20				
10	Column 5	20	60	57	0	Protein 5	30	20	50	40				
11										Enter Height here				
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														
32														

Process Areas | Equip Information | Column Info | SplitCombine | PSD Data | Utilities | Main Activities | M

Split/Combine Tab

- The splitting and combining of each batch is determined either based on the fermentation titre and is therefore calculated, or it is based on a user defined ratio.
  The method of determination is set in the General Parameters table. Combine ratio is calculated on a similar basis
- The split and combine ratios are based on titre +20%, the upper range of the titre fluctuation
- The locations for split and combine are also user defined
- For Split Ratio based on Titre, following eqn is used: *Ratio = (Titre * FermenterVolume * RecoveryYield) / (dbcCapture*Capture_Volume), rounded up to the nearest even integer*
- For Comb Ratio based on Titre, following eqn is used: *Ratio = (Titre * FermenterVolume * (RecoveryYield + Yield of Steps before Column)) / (dbcCol*Col_Volume), rounded up to the nearest even integer*

Split determination method: **Titre Based, Limited** (set in General Parameters)

		No Limit Titre Based	Limited Titre based	User Defined	Location	Split/Comb	Discarded
P1	Split Ratio	4	4	8	13	4	0
	Combine Ratio	4	2	4	17	2	
P2	Split Ratio	12	8	6	13	8	4
	Combine Ratio	6	4	2	17	4	
P3	Split Ratio	8	8	6	13	8	0
	Combine Ratio	4	4	3	17	4	

--> **NOTE!** Lots discarded are not actually thrown away. This means that the column is loaded beyond its binding capacity and therefore the volumetric loads is always the same. Due to the overloading, some product doesn't bind and flows through.

Calculation of combine ratio for unlimited cycles: 4, 11, 6

NOTE This table contains calculations based on data entered 5.1020408

		P1			P2			P3			Batch Split/Combined Lots			Cycles		
ActivityID	Activity	Lot Discard	Split	Combine	Lot Discard	Split	Combine	Lot Discard	Split	Combine	P1	P2	P3	P1	P2	P3
1			1	1		1	1		1	1	1	1	1	1	1	1
2			1	1		1	1		1	1	1	1	1	1	1	1
3			1	1		1	1		1	1	1	1	1	1	1	1
4			1	1		1	1		1	1	1	1	1	1	1	1
5			1	1		1	1		1	1	1	1	1	1	1	1
6			1	1		1	1		1	1	1	1	1	1	1	1
7			1	1		1	1		1	1	1	1	1	1	1	1
8			1	1		1	1		1	1	1	1	1	1	1	1
9			1	1		1	1		1	1	1	1	1	1	1	1
10			1	1		1	1		1	1	1	1	1	1	1	1
11			1	1		1	1		1	1	1	1	1	1	1	1
12			1	1		1	1		1	1	1	1	1	1	1	1
13		0	4	1	4	8	1	0	8	1	4	8	8	4	8	8
14			1	1		1	1		1	1	4	8	8	4	8	8
15			1	1		1	1		1	1	4	8	8	4	8	8
16			1	1		1	1		1	1	4	8	8	4	8	8
17			1	2		1	4		1	4	2	2	2	2	4	4

Process Areas | Equip Information | Column Info | **SplitCombine** | PSD Data | Utilities | Main Activities | M

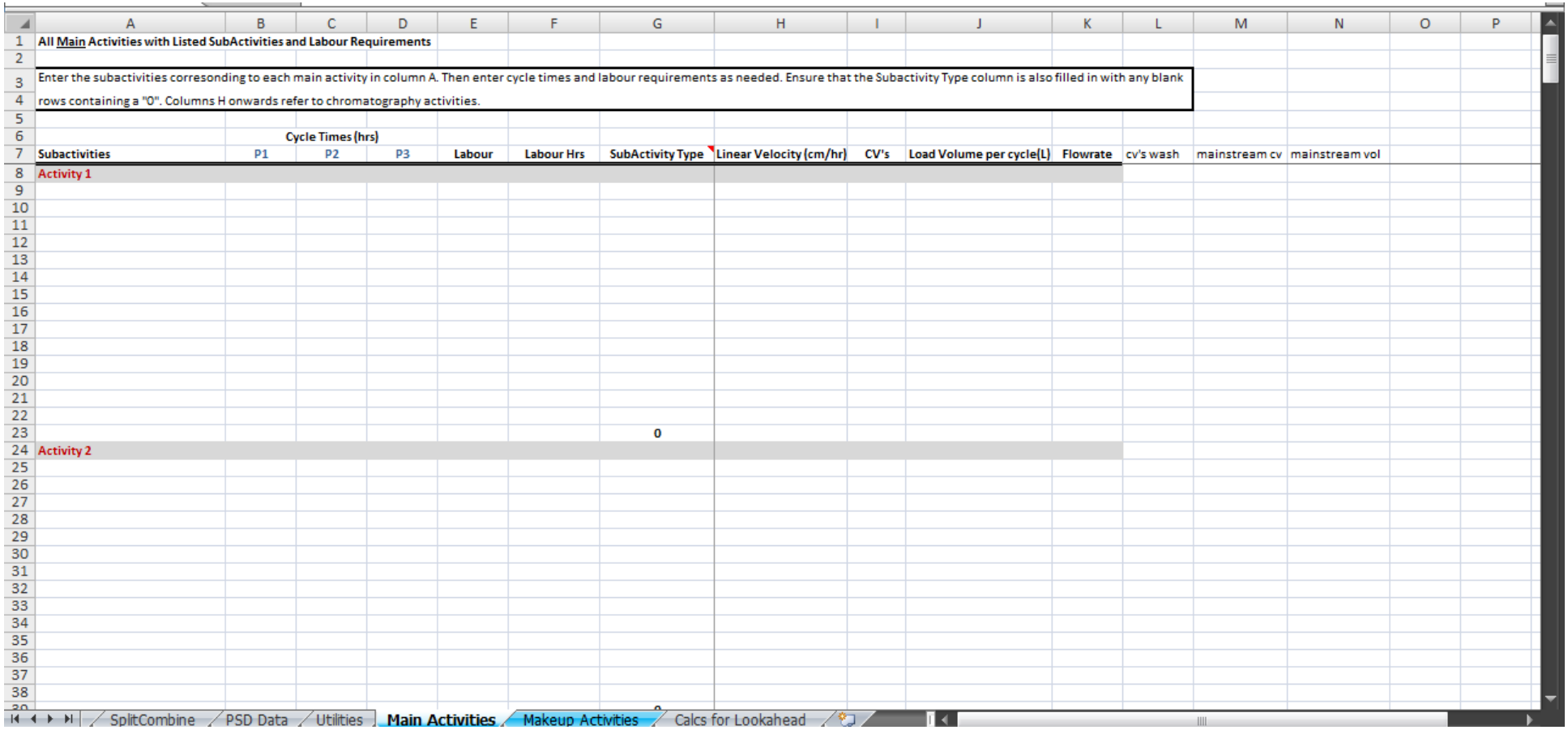
PSD Data Tab

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Equipment	Equipment Type	Clean Expiry	Max Rinse Only Cycles	CIP Total WFI-AMB Quantity	CIP Total WFI-AMB Flowrate	CIP Cycle Time	CIP Total WFI-Hot Quantity	CIP Total WFI-Hot Flowrate	Rinse Only Cycle Time	Rinse Only WFI Quantity	Rinse Only WFI Flowrate	SIP Cycle Time	SIP CSTM Quantity	SIP CSTM Flowrate	Makeup WFI Quantity	Makeup WFI Flowrate
2																	
3																	
4																	
5																	
6																	
7																	
8																	

Utilities Tab

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Utility	Description	Max Capacity (L)	Fill Factor Initial Flush	Fill Factor Final Flush								
2	CSTM	clean steam for SIP											
3	WFI_AMB	for media/buffer makeup											
4	WFI_HOT	for CIP											
5													
6													
7													
8													
9													

Main Activities Tab



All Main Activities with Listed SubActivities and Labour Requirements

Enter the subactivities corresonding to each main activity in column A. Then enter cycle times and labour requirements as needed. Ensure that the Subactivity Type column is also filled in with any blank rows containing a "0". Columns H onwards refer to chromatography activities.

		Cycle Times (hrs)												
Subactivities	P1	P2	P3	Labour	Labour Hrs	SubActivity Type	Linear Velocity (cm/hr)	CV's	Load Volume per cycle(L)	Flowrate	cv's wash	mainstream cv	mainstream vol	
Activity 1														
						0								
Activity 2														

SplitCombine | PSD Data | Utilities | Main Activities | Makeup Activities | Calcs for Lookahead

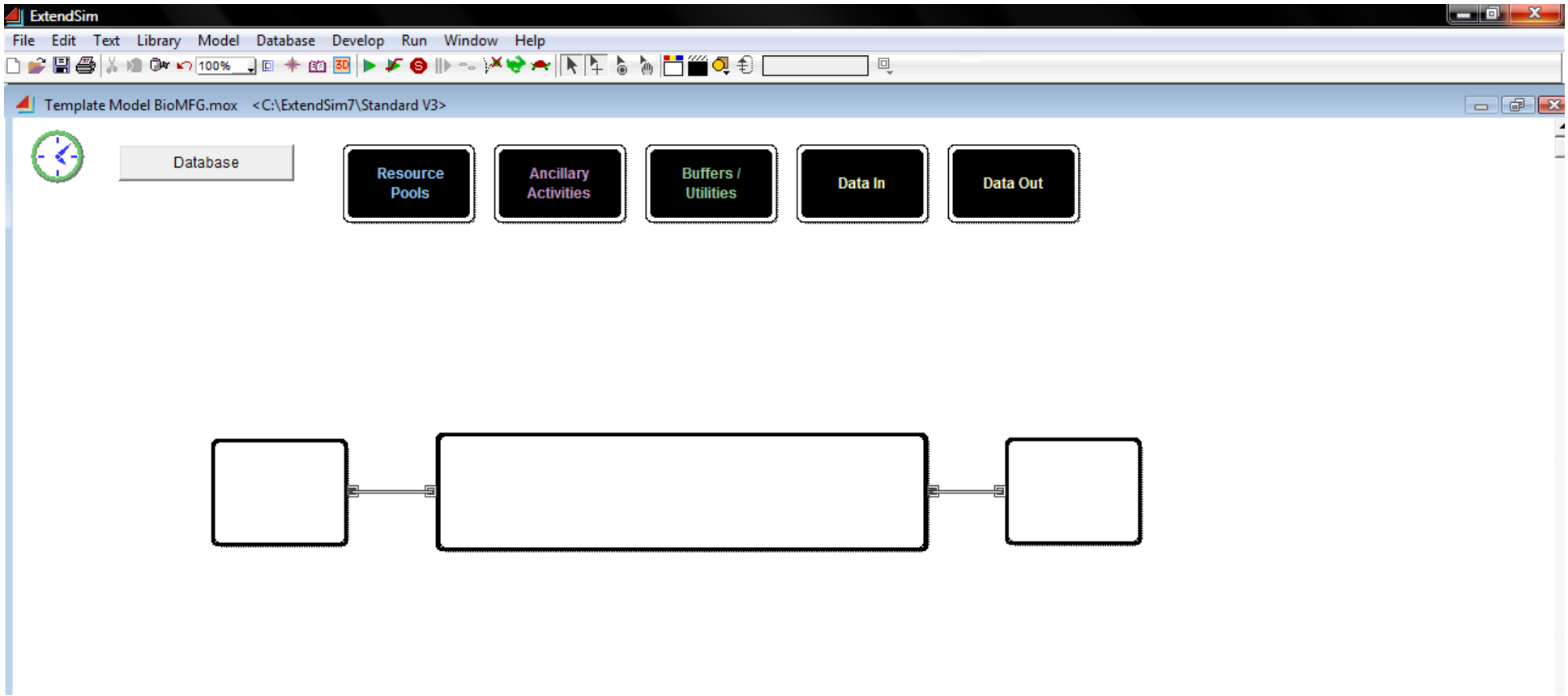
Calcs for Lookahead Tab

	A	B	C	D	E	F	G	H	I	J
1	NOTE This worksheet contains calculations based on data entered elsewhere. Do not overwrite									
2										
3		Preparation Cycle Times			Run Cycle Times			Cumulative Cycle Times exc Inoc		
4	Activity/Equipment	P1	P2	P3	P1	P2	P3	P1	P2	P3
5	Activity 1	0	0	0	0	0	0	0	0	0
6	Activity 2	0	0	0	0	0	0	0	0	0
7	Activity 3	0	0	0	0	0	0	0	0	0
8	Activity 4	0	0	0	0	0	0	0	0	0
9	Activity 5	0	0	0	0	0	0	0	0	0
10	Activity 6	0	0	0	0	0	0	0	0	0
11	Activity 7	0	0	0	0	0	0	0	0	0
12	Activity 8	0	0	0	0	0	0	0	0	0
13	Activity 9	0	0	0	0	0	0	0	0	0
14	Activity 10	0	0	0	0	0	0	0	0	0
15	Activity 11	0	0	0	0	0	0	0	0	0
16	Activity 12	0	0	0	0	0	0	0	0	0
17	Activity 13	0	0	0	0	0	0	0	0	0
18	Activity 14	0	0	0	0	0	0	0	0	0
19	Activity 15	0	0	0	0	0	0	0	0	0
20	Activity 16	0	0	0	0	0	0	0	0	0
21	Activity 17	0	0	0	0	0	0	0	0	0
22	Activity 18	0	0	0	0	0	0	0	0	0
23	Activity 19	0	0	0	0	0	0	0	0	0
24	Activity 20	0	0	0	0	0	0	0	0	0
25	Activity 21	0	0	0	0	0	0	0	0	0
26	Activity 22	0	0	0	0	0	0	0	0	0
27	Activity 23	0	0	0	0	0	0	0	0	0
28	Activity 24	0	0	0	0	0	0	0	0	0
29	Activity 25	0	0	0	0	0	0	0	0	0
30										
31										
32										

SplitCombine | PSD Data | Utilities | Main Activities | Makeup Activities | Calcs for Lookahead

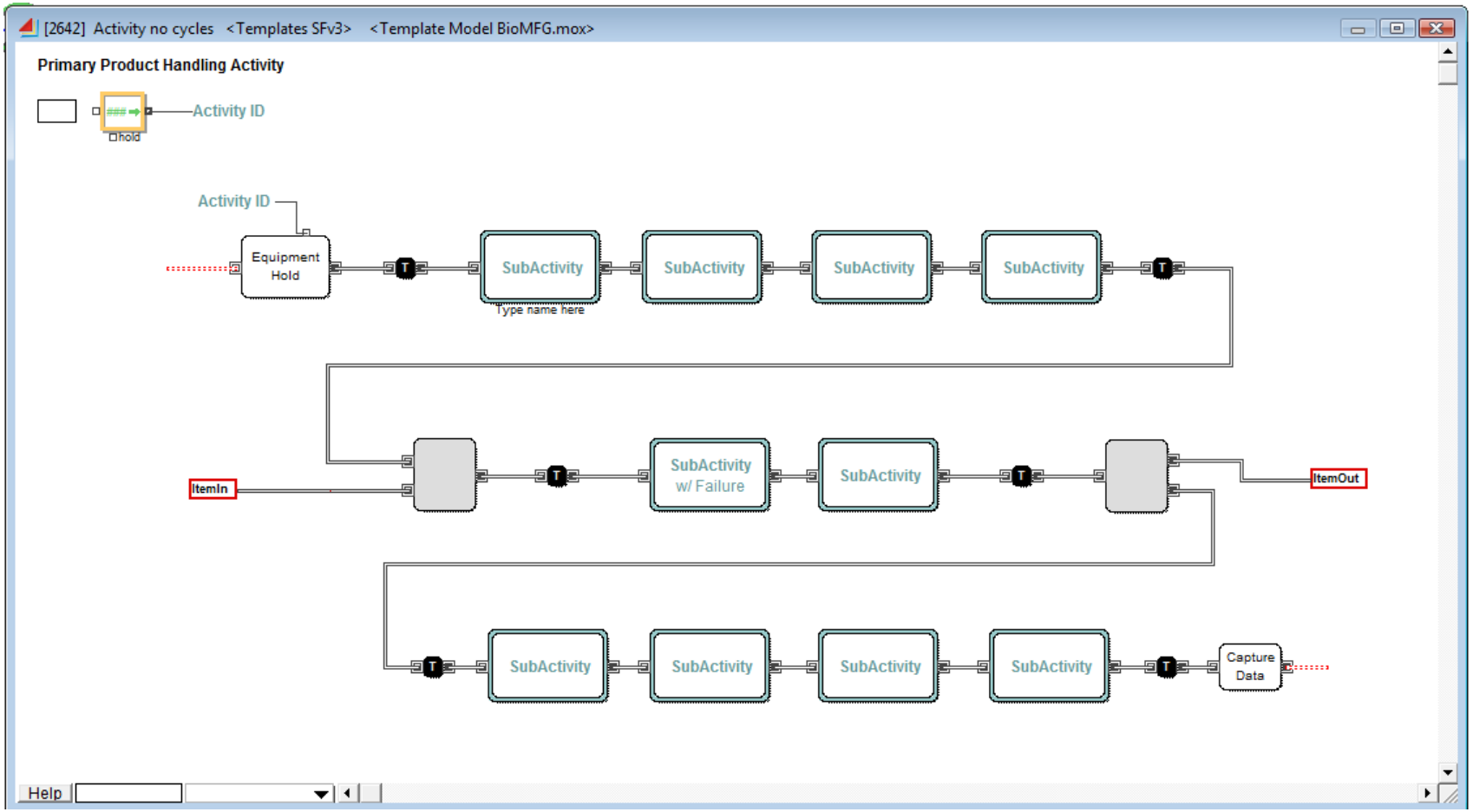
Appendix A2 Standard Framework Version 3: Library Templates

Model Template Top Level



Activities Library

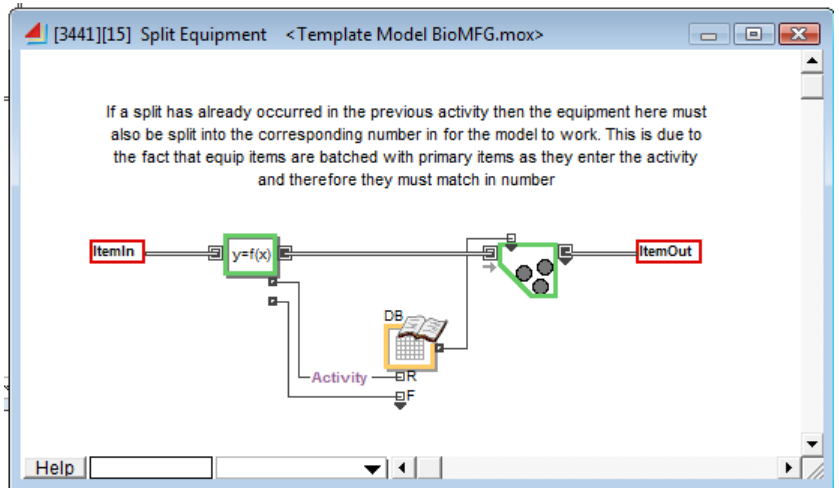
Activity No Cycles

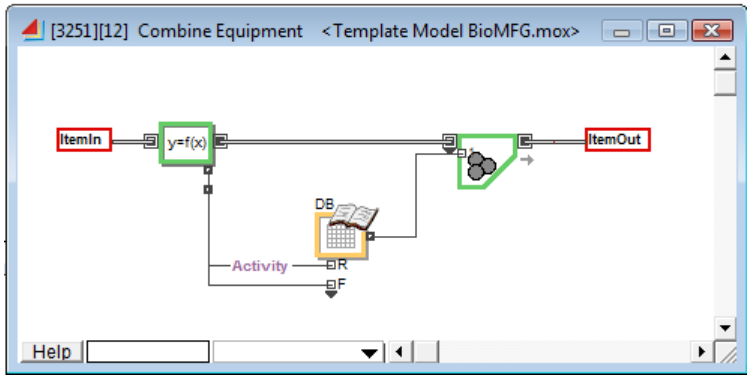


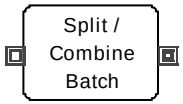
Activity with Cycles

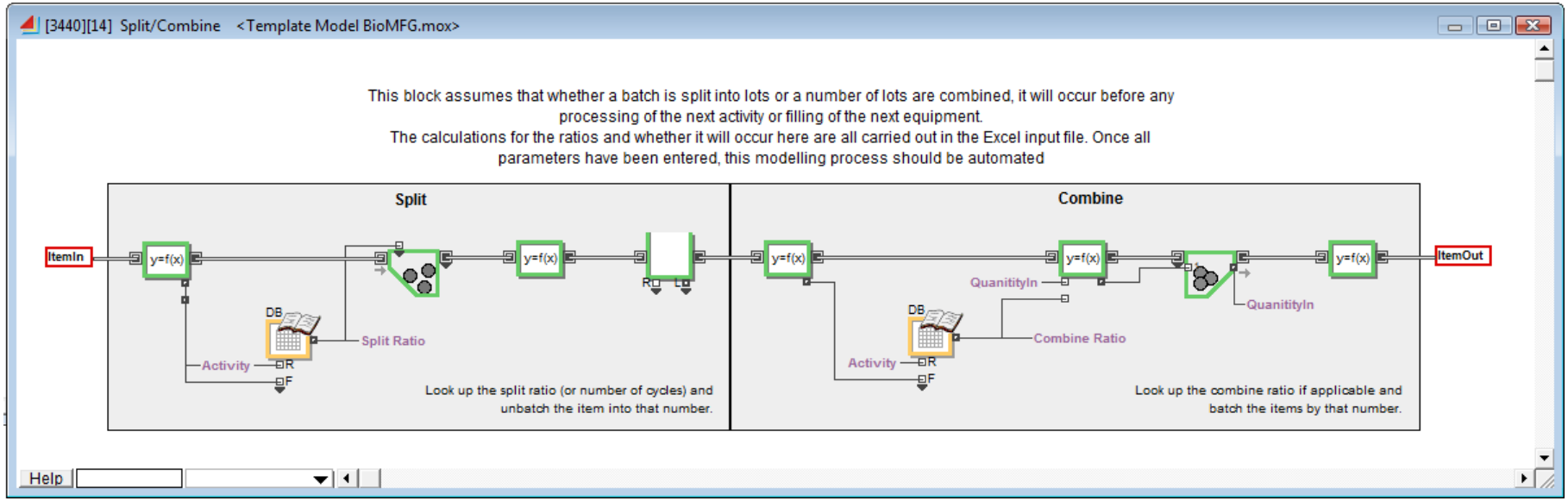
The block is identical to the Activity no Cycles block, but only at the top most level.

At the next level however, it contains the following, ‘Split Equip’, ‘Combine Equip’ and ‘Split/Combine Batch’

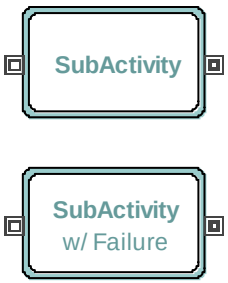


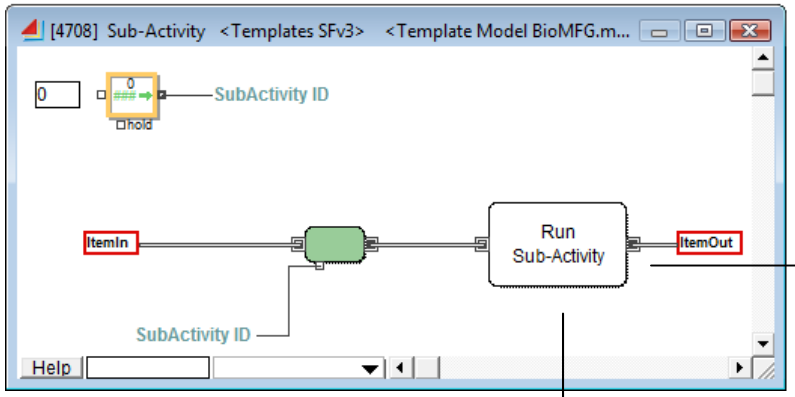



Split /
Combine
Batch

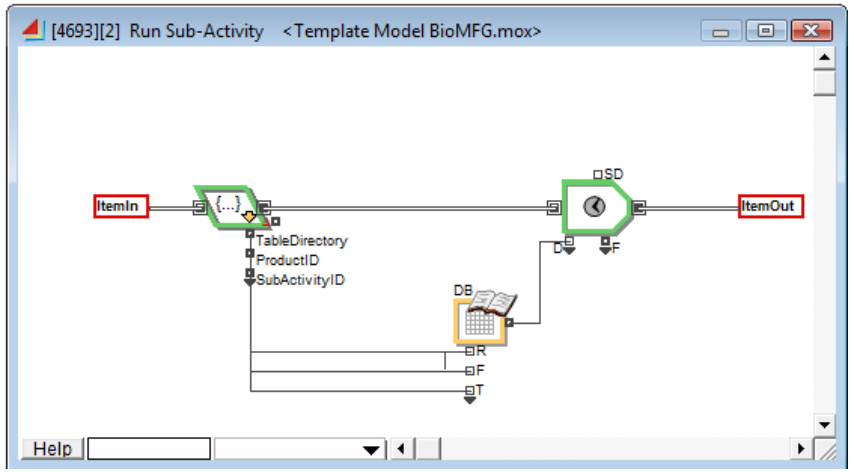

[3440][14] Split/Combine <Template Model BioMFG.mox>
This block assumes that whether a batch is split into lots or a number of lots are combined, it will occur before any processing of the next activity or filling of the next equipment.
The calculations for the ratios and whether it will occur here are all carried out in the Excel input file. Once all parameters have been entered, this modelling process should be automated
Split
Combine
ItemIn
ItemOut
y=f(x)
DB
Activity
R
F
Split Ratio
Look up the split ratio (or number of cycles) and unbatch the item into that number.
QuanitityIn
Combine Ratio
Look up the combine ratio if applicable and batch the items by that number.
Help

SubActivity

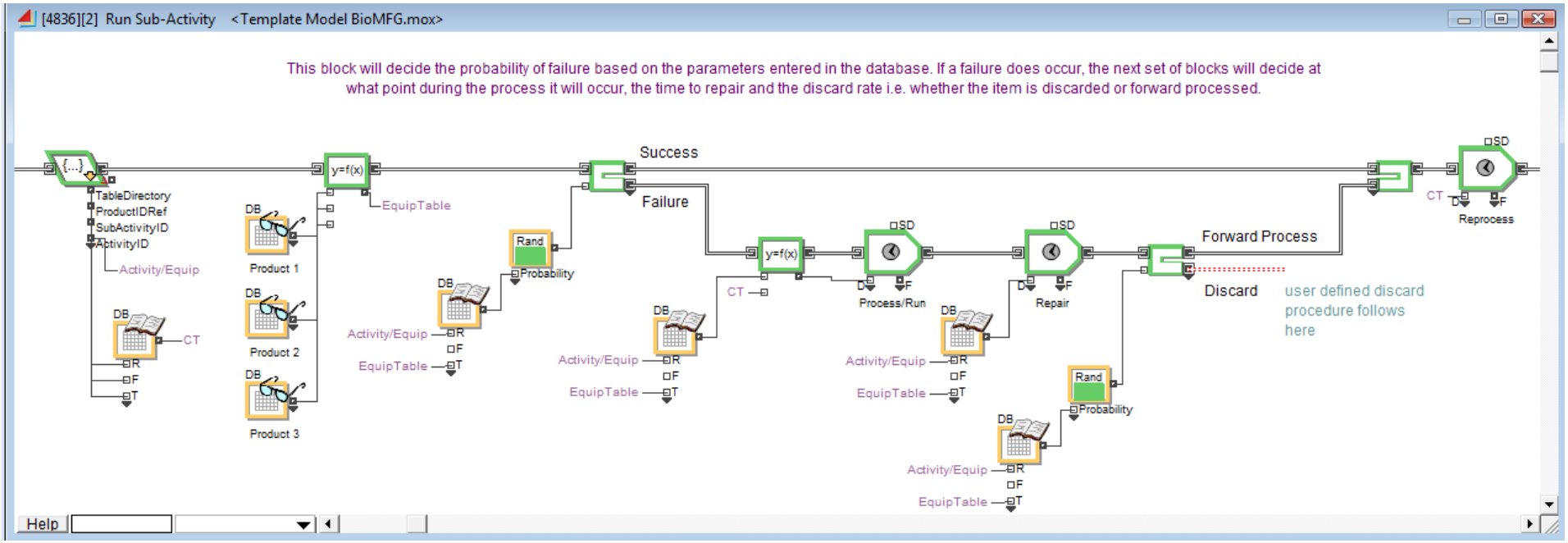




Without Failure

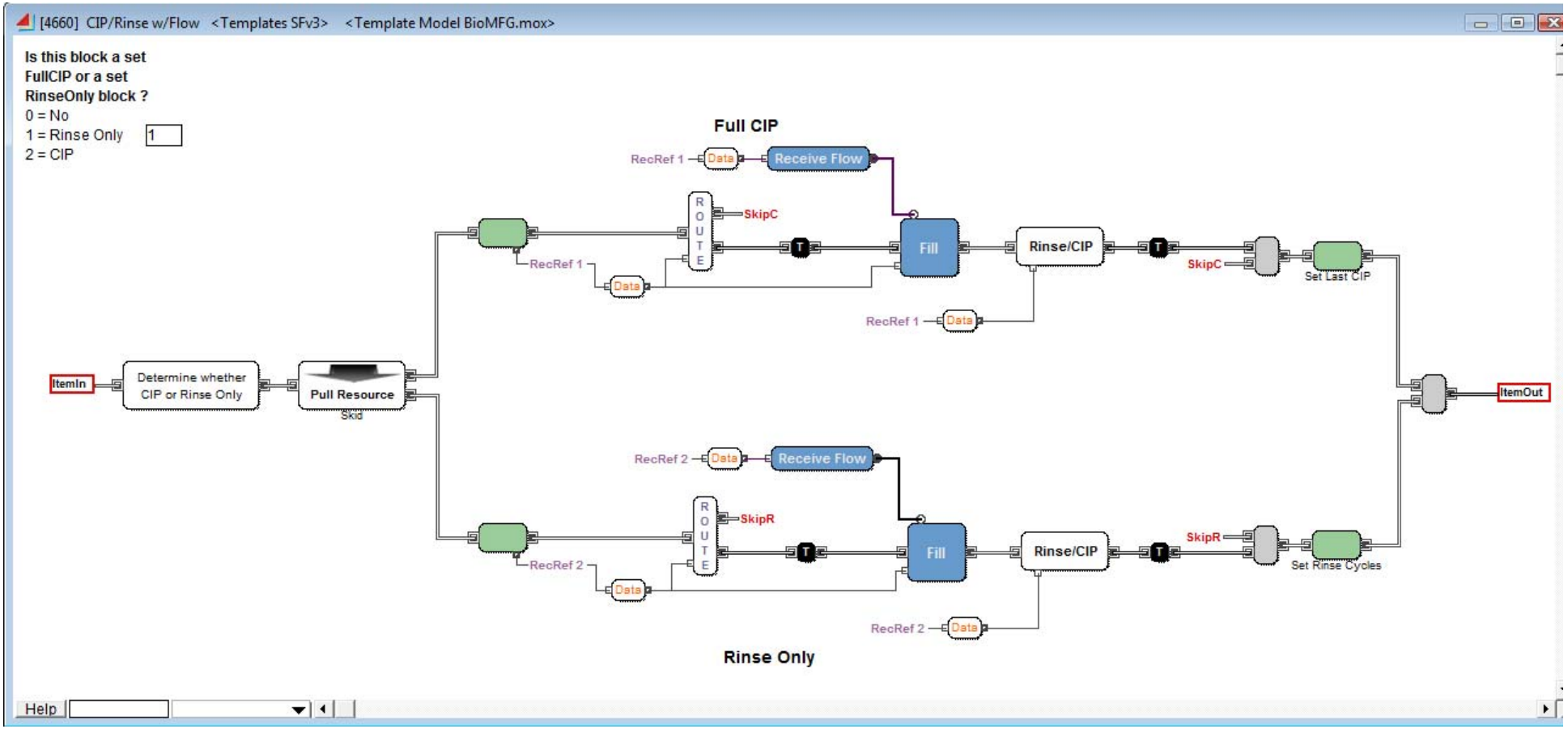


With Failure



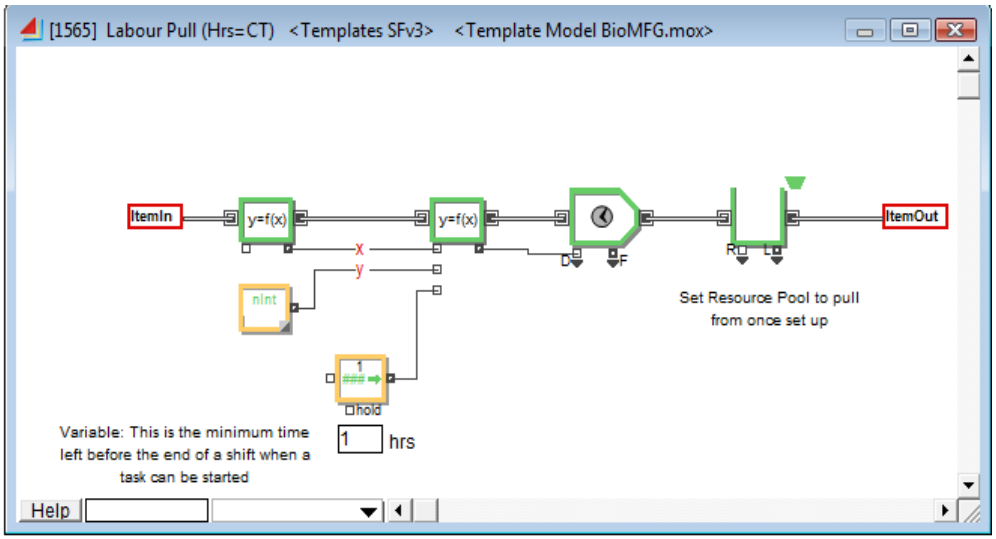
CIP


CIP/
Rinse Only

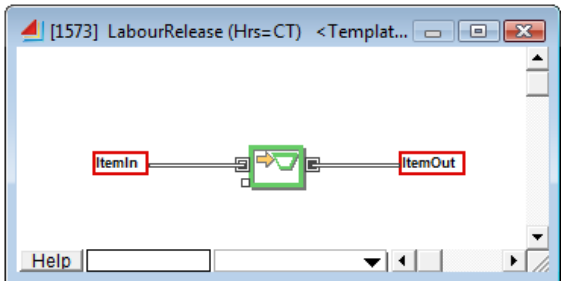

[4660] CIP/Rinse w/Flow <Templates SFv3> <Template Model BioMFG.mox>
Is this block a set
FullCIP or a set
RinseOnly block ?
0 = No
1 = Rinse Only
2 = CIP
1
Full CIP
RecRef 1
Data
Receive Flow
ROUTE
SkipC
Fill
Rinse/CIP
SkipC
Set Last CIP
RecRef 1
Data
RecRef 1
Data
ItemIn
Determine whether
CIP or Rinse Only
Pull Resource
Skid
ItemOut
RecRef 2
Data
Receive Flow
ROUTE
SkipR
Fill
Rinse/CIP
SkipR
Set Rinse Cycles
RecRef 2
Data
RecRef 2
Data
Rinse Only
Help

Labour Library

Labour Pull (Hr=CT)

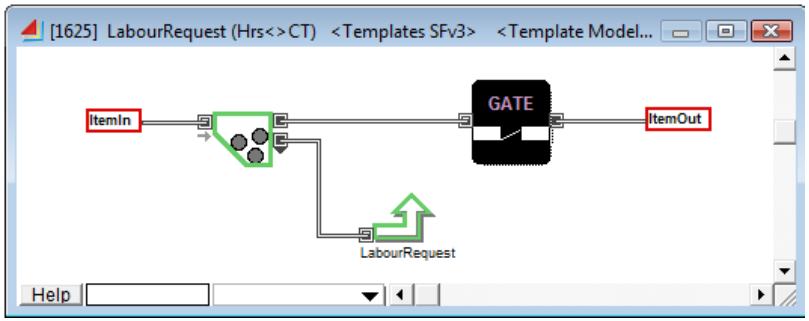


Labour Release (Hr=CT)

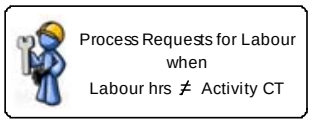


Labour Request (Hr<>CT)

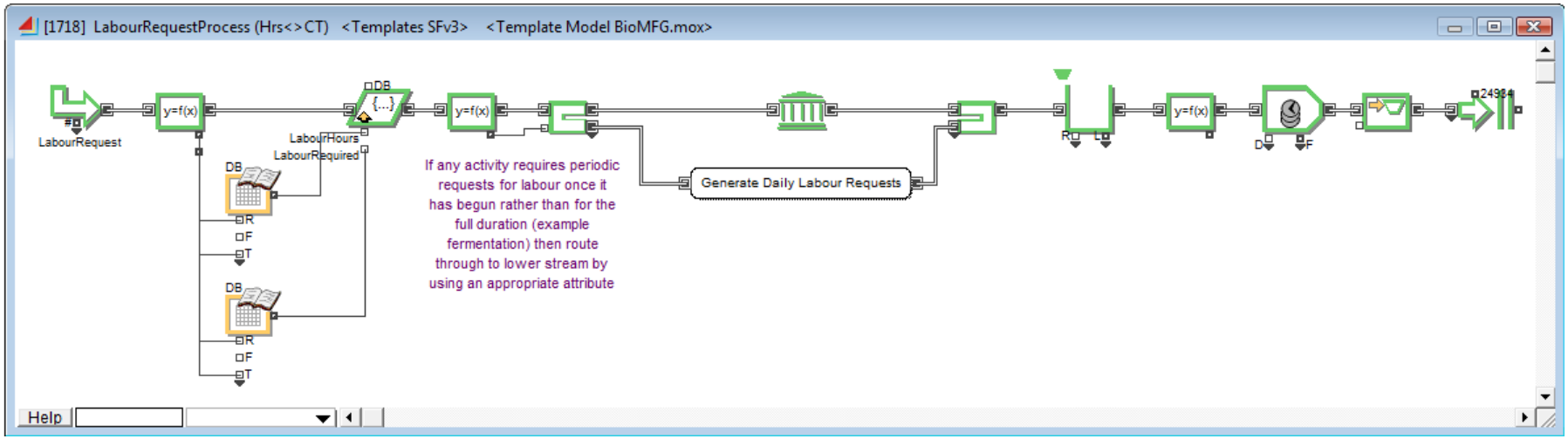
Request
Labour



Labour Request Process (Hr<>CT)

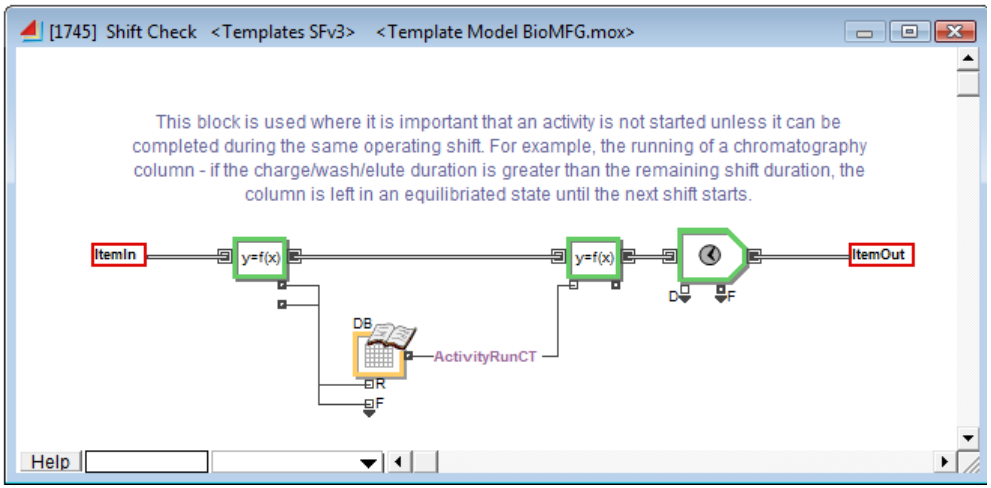


Process Requests for Labour
when
Labour hrs ≠ Activity CT



Shift Check

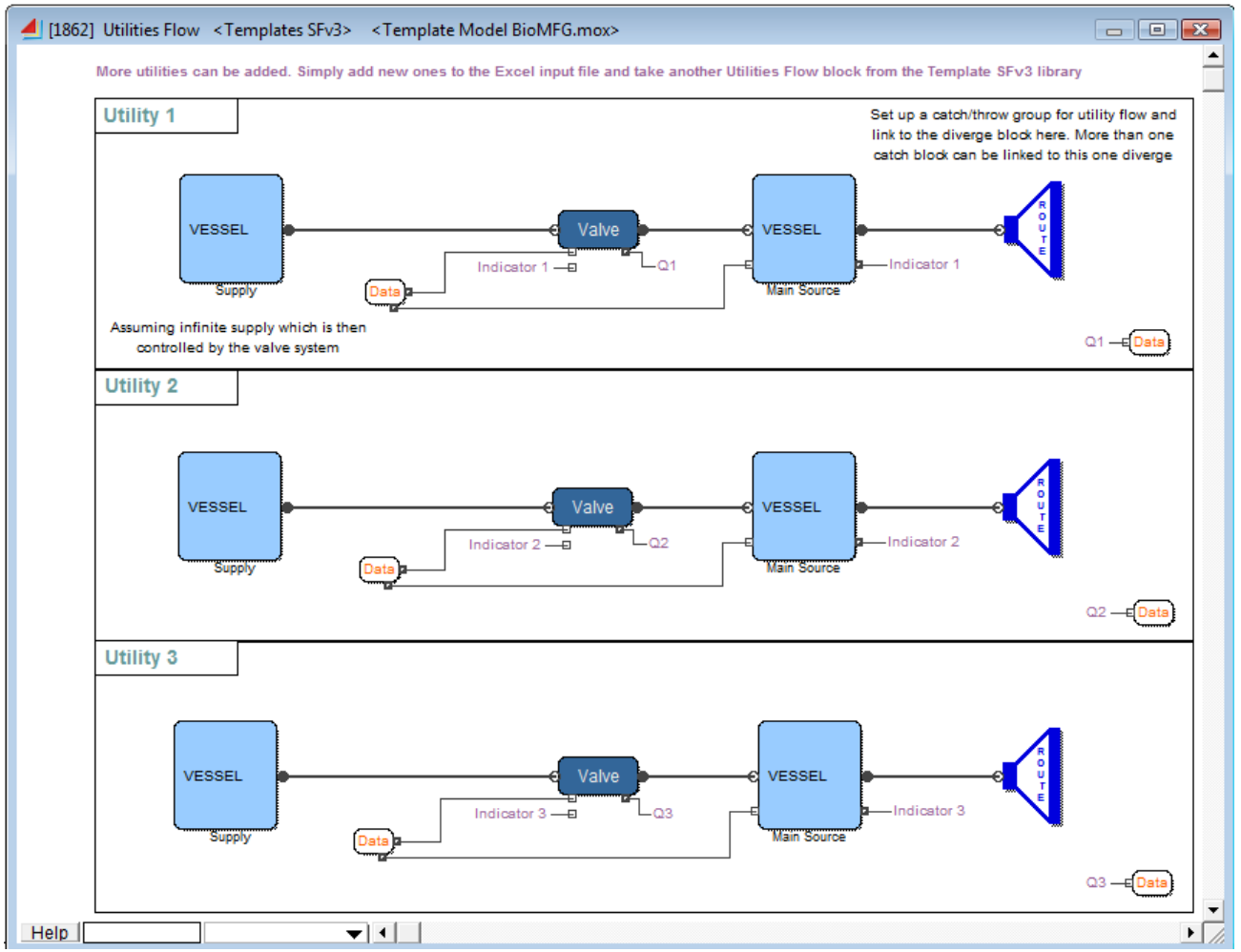

Shift Check


[1745] Shift Check <Templates SFv3> <Template Model BioMFG.mox>
This block is used where it is important that an activity is not started unless it can be completed during the same operating shift. For example, the running of a chromatography column - if the charge/wash/elute duration is greater than the remaining shift duration, the column is left in an equilibriated state until the next shift starts.
ItemIn
y=f(x)
y=f(x)
ItemOut
D
F
DB
ActivityRunCT
R
F
Help

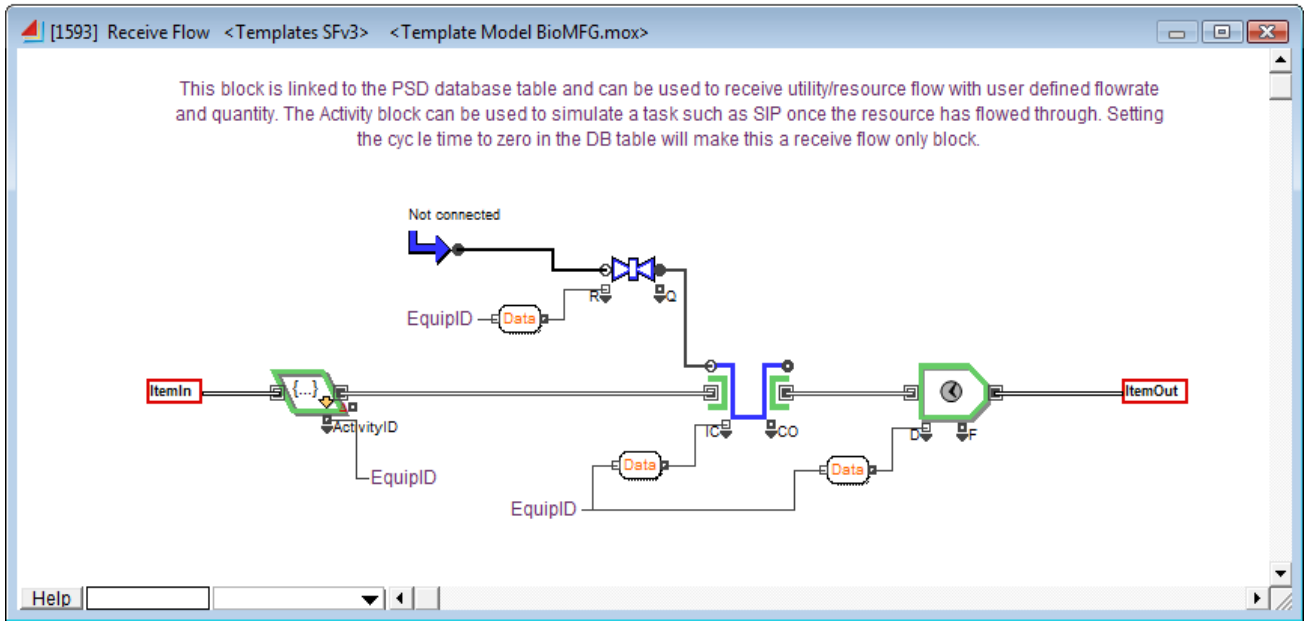
Resources Library

Utilities

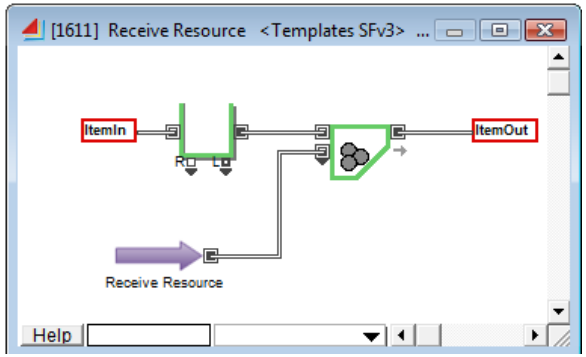
Utilities



Receive Flow

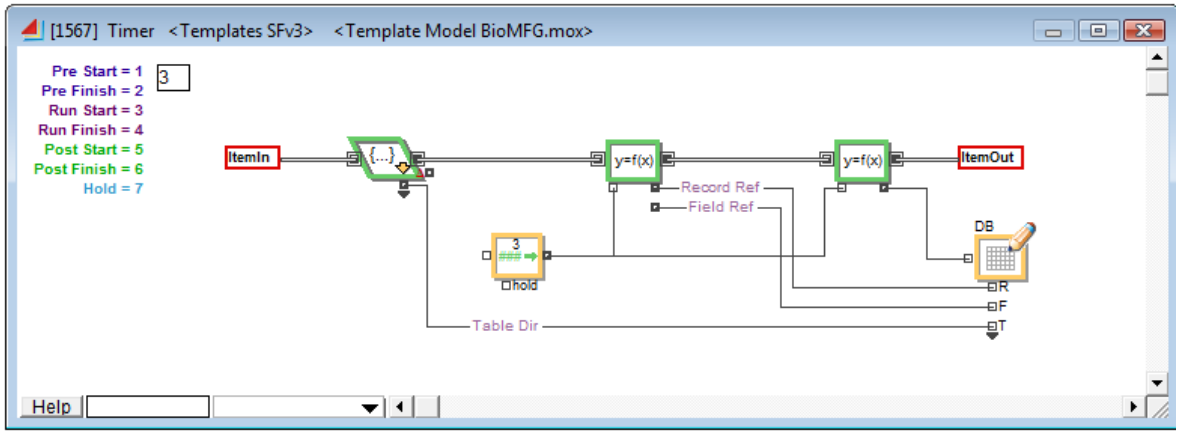


Receive Resource

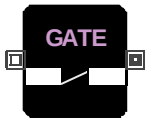


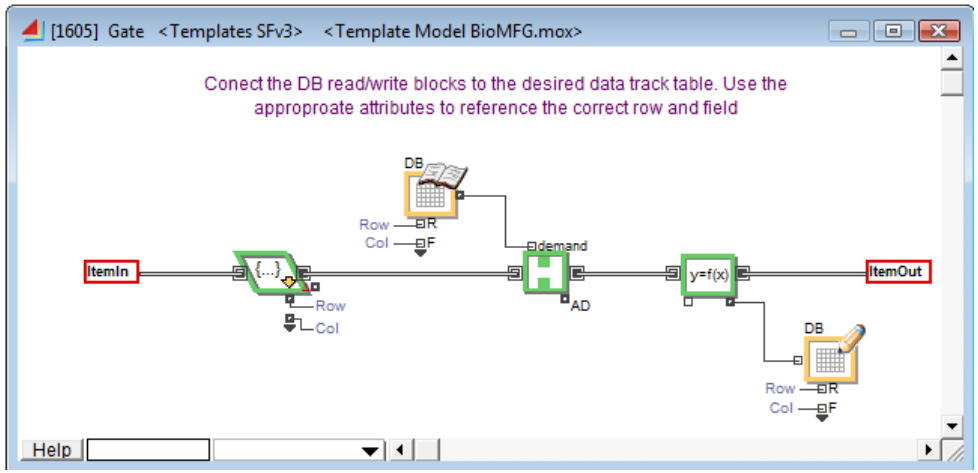
Logic Library

Timer



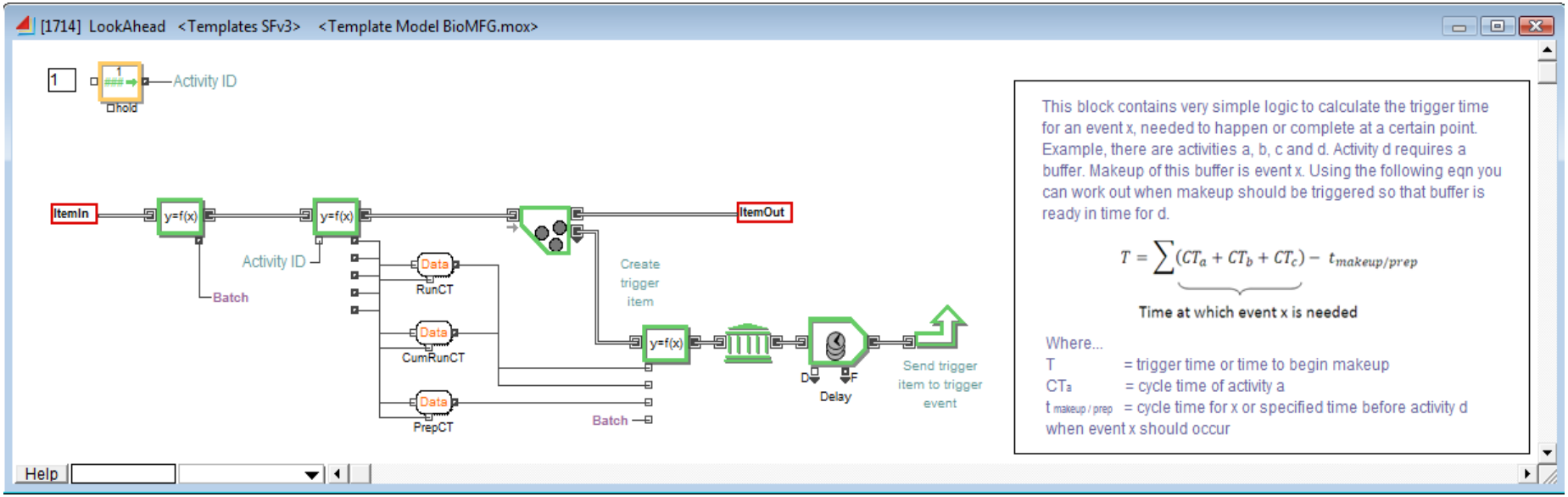
Gate



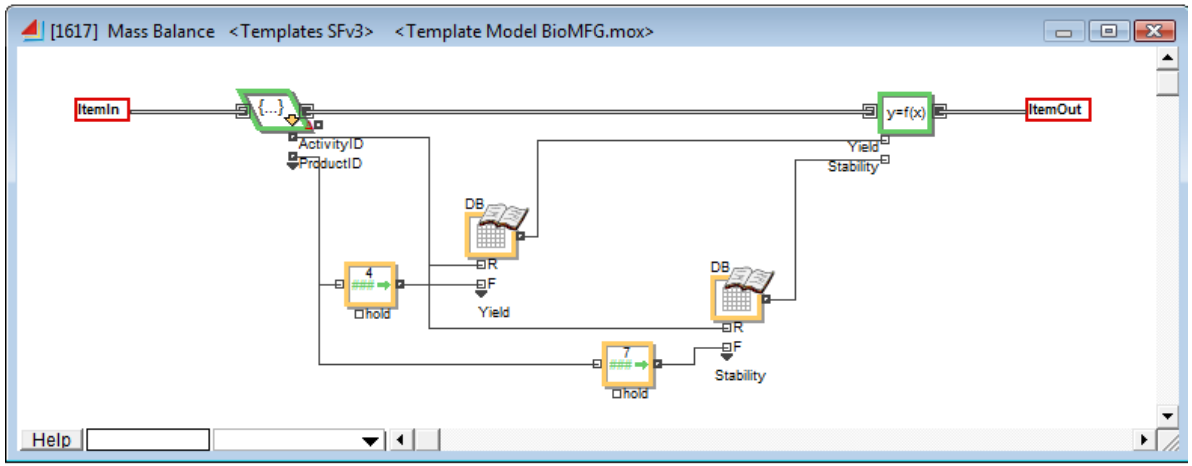


Lookahead

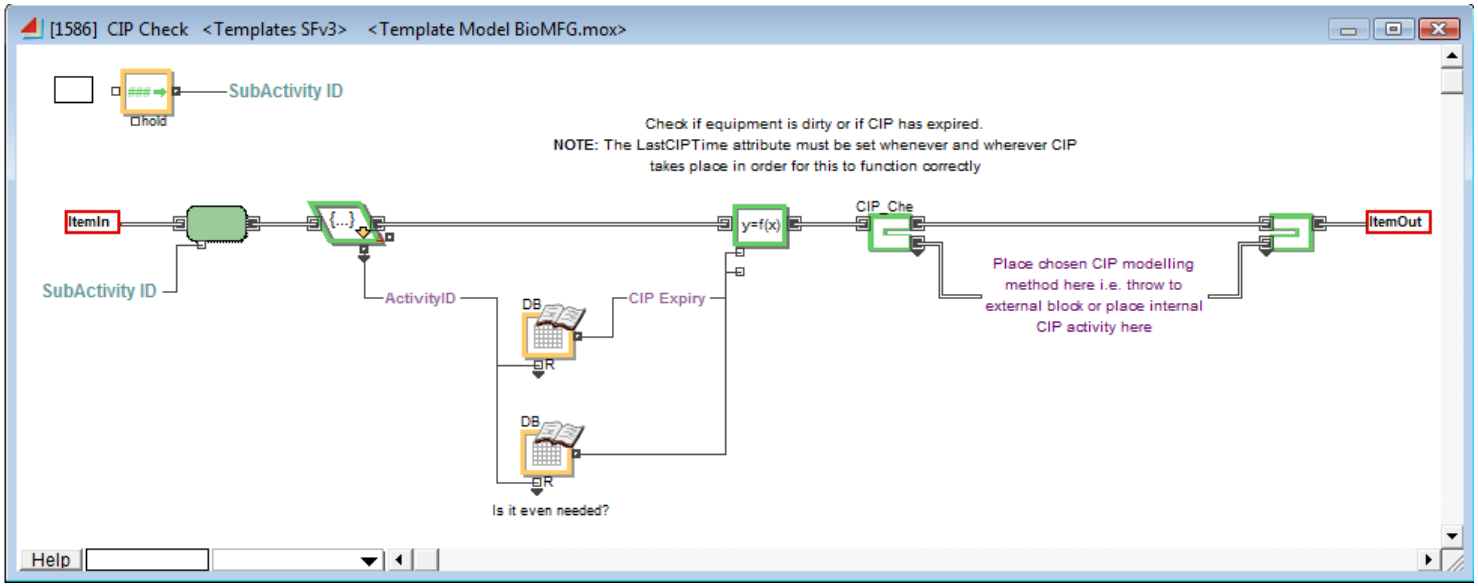

LookAhead


[1714] LookAhead <Templates SFv3> <Template Model BioMFG.mox>
1
Activity ID
hold
ItemIn
y=f(x)
y=f(x)
ItemOut
Activity ID
Batch
Data
RunCT
Data
CumRunCT
Data
PrepCT
Create trigger item
y=f(x)
D
F
Delay
Send trigger item to trigger event
Batch
Help
This block contains very simple logic to calculate the trigger time for an event x, needed to happen or complete at a certain point. Example, there are activities a, b, c and d. Activity d requires a buffer. Makeup of this buffer is event x. Using the following eqn you can work out when makeup should be triggered so that buffer is ready in time for d.
$T = \sum(CT_a + CT_b + CT_c) - t_{makeup/prep}$
Time at which event x is needed
Where...
T = trigger time or time to begin makeup
CTa = cycle time of activity a
t makeup / prep = cycle time for x or specified time before activity d when event x should occur

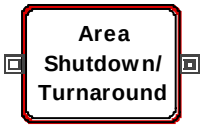
Mass Balance

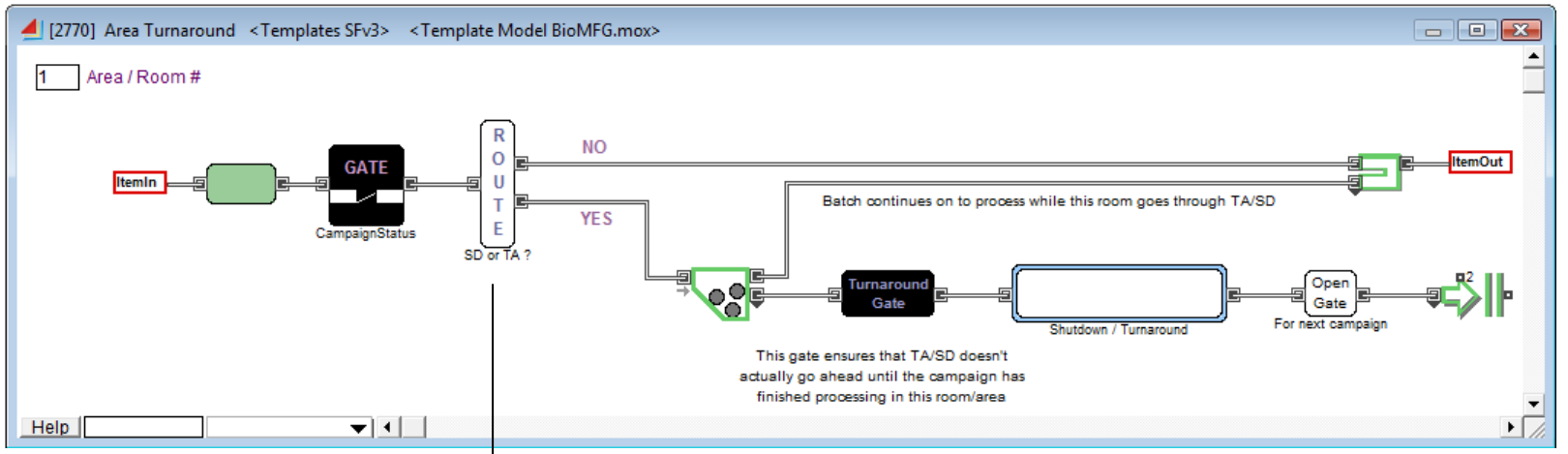


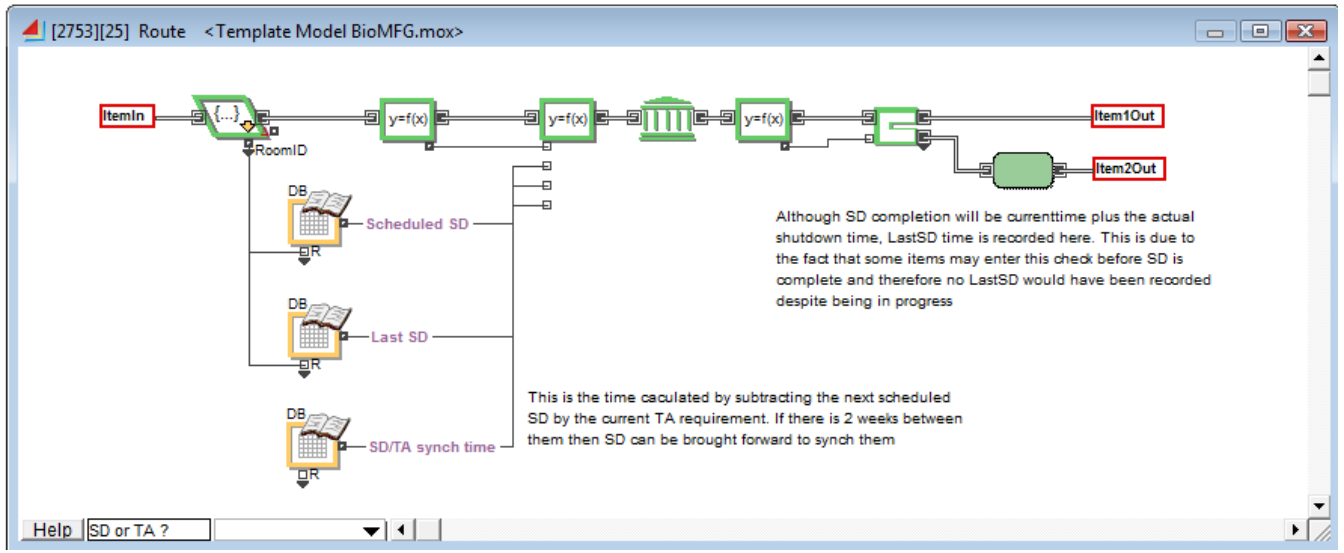
CIP Check



Area Turnaround







APPENDIX B

Data for Chapter 4

Appendix B.1 Rationale behind construction of BioSynT model

MODELLING THE ELEMENTS

The following looks at the different system and model elements, describing the reasons for modelling them as they were modelled.

Activities

CIP

CIP is an activity associated with the chromatography columns. Within the scope of this problem there is no requirement for the modelling of CIP component make-up. Cleaning procedures only correspond to equipment used either within the main process stream or for ancillary activities such as buffer preparation and waste disposal. There are two ways in which CIP can be modelled and a number of reasons for choosing one method over the other. ---+

+

Method 1: As part of the main process chain logic. This means that CIP is placed wherever the activity is required. This method means that there may be multiple (depending on the number of clean occurrences) CIP activities and is used when the equipment are modelled as the main block constructs (role decomposition).

Method 2: The CIP is represented in one location within the model which is externalised from the main process chain. This method is used when there is functional decomposition. When cleaning is required the entity to be cleaned is sent to that clean location and then returned to the process stream. This is the chosen method for this case study due to the following reasons. Firstly it reduces the number of building blocks (visible complexity) and thus allows for a degree of reduction in

the model run time. Secondly it creates a more intuitive model in the case as it allows for visual tracking of item movement through clean. Finally if a generic CIP is built, it will contribute greatly to the maximisation of model reusability and sustainability.

Preparation of Buffers

These activities are buffer preparation and each step of this preparation process will be modelled as an individual task. Preparation of buffers, similarly to CIP can be modelled as part of the product handling chain and therefore have multiple occurrences or can be modelled externally. The latter is the chosen method due to block use efficiency and clarity of model since there are a significant number of sub-tasks associated with the preparation of each buffer.

Entities

Entities are those objects in the modelling environment which represent people or items (inc. batches and equipment) in the real system. The *input items* to this system are containers of frozen intermediate from the anion exchange chromatography in the front end process. These intermediates are simply batches and are the primary items. The generation of these items is not based on a schedule due to the fact that they do not directly come from the FrontEnd but are thawed as and when needed.

Resources

There are two types of resource in this system, consumable and non consumable. The former are the buffers prepared at various stages before use. The non consumable resources are the equipment and labour. All these resources can be modelled in two different ways:

Method 1: Modelled as items. This is beneficial when it is necessary to visually track the resource's movements, when the resource and primary item need to be paired for a section of the model, and/or the resource carries its own attributes. Furthermore, importantly, it is necessary to use this method when the resource has activities associated with it which are independent of the primary item activities. The resource item can either be generated then discarded or be managed via a holding block where it sits until required. The advantage of this method is that an item can be turned into a volume flow at any point. This is useful in situations where a transfer activity over a period of time must be represented, particularly when there are flow restrictions.

Method 2: Held in resource pools and pulled when needed. Resources held in pools are not actual items which can move through the model; instead the number available for that resource is changed every time it is pulled or released, telling the model its availability. This method should be used when the resource itself does not have any activities associated with it that are independent of the primary entities. Therefore as the primary item enters the activity it will pull the resource and carry it through the series of activities it must go through before releasing it back to the resource pool.
One pool may be used to hold all similar resources if individual utilisation data is not required or it is calculated elsewhere and/or usage is based on the same shift/rule or this is determined elsewhere.

Non-consumable resources such as equipment should be modelled using method 1 because they have independent activities and should be tracked for debugging and general visibility purposes.
Due to the fact that labour does not usually have associated with it activities independent of the primary item's there would be little added benefit to modelling them as physical items within the model. Therefore it is best to use method 2. The constraints due to labour shortage are modelled much in the same way as method 1 i.e. if the labour is not available the item requesting it will simply sit in a queue. However in addition the resource pool gives the extra option of labour allocation based on first come first served or a given priority.

Consumable resources such as water or buffer chemicals should be held in resource pools using method 2 as they do not loop back to be used again (unless recycling of such materials is part of the system being modelled).

Physical Layout and Decomposition

The physical layout of a model is the position of blocks on the workspace and what they represent or the decomposition of the model. Discrete event simulation modelling in Extend involves hierarchal levels containing blocks. For an intuitive model it is necessary to reduce complexity at each level. The IDEF0 standard describes sub levels of a model as Child Diagrams (the top level being Parent Diagram), whereby functions are decomposed into their sub-functions. According to the standard, the number of child boxes on each child diagram should be limited to 3-6. The standard describes the following decomposition types:

Functional Decomposition – breaks down activities according to what is done. This is the most common decomposition strategy
Role Decomposition – breaks down things according to who does what. This definition should be extended to also mean according to what does what (equipment based constructs)
Lifecycle Decomposition – breaks down a system first by the phases of activity

Functionalities found in reality should be represented in the model for both visualization purposes and for resource management by using the IDEF0 decomposition criteria.

The BioSynT process lends itself to Functional Decomposition and therefore the main hierarchical blocks will represent the major activities, in this case the purification, pooling and drying steps, which form the primary process stream.

Main Block Construct and Parallel Activities

As the main block construct will consist of chromatography columns and pooling tanks the second level of the model will have physical layout similar to that shown in Figure 4.1 in Chapter 4. For an intuitive model, functions should also ideally be hierarchically modelled together. E.g. if two equipments / activities perform the same function, there should only be one hierarchical block representing that function to contain them. In the real system the initial reverse phase chromatography activity consists of two columns whereas only one block is used to represent it.

Item Transfer

The IDEF0 standard considers the way in which boxes and arrows are used to represent objects and flow; That is, the sequence of events flow from left to right with arrows used to connect boxes which represent the object/activity being modelled. Since this convention is simple to understand and quite commonly used in flow diagrams, it will be adopted here, in order to create a more intuitive model.

Item Flow Control

The rules and constraints within a model make it necessary to control the flow of items until certain conditions are met. The main and possible best was of controlling flow is to use gates. The function of these is to check whether the required conditions have been met and to release the item accordingly.

Primary Items Generated

The main items to be generated in the model are batches. Since one possible capacity management question considers multi-products, the database can be used to generate these based on a schedule. Then the products to be generated in each case (there can be combinations of products or only one as in this case study) can be chosen by the user and only those generated.

Metrics

Having described the possible capacity management scenarios, it is now necessary to determine what the model should be capable of measuring in order to be robust enough to be used for any one or combination of questions in addition to answering the question of 'how fast can this process be run?'. Therefore the following describes the required measuring capabilities of the model.

Cycle Times: In order to track the overall completion cycle time and the utilisation profiles of all the activities or equipment, it is necessary to record the times at which events occur. The data when compiled will then make it possible to create Gantt charts for activities.

Throughput: This is both overall and for each step of the process and can be tracked by simply using looking at the number of items which exited the model. This is the case due to the fact that each item represents a batch; therefore the number of items which pass through a section of the model will represent the number of batches processed.

Equipment utilisation: Tracking the utilisation of equipment is not simply the case of recording the percentage of time not held in its pallet. When a system is as complex as in this case with many sub activities, resources and constraints, it is far more useful to track the times which the equipment takes to complete these activities and also the how long it is sat waiting for a resource to become available. The best way to do this is to assign time attributes which can be sent to Excel and used to create Gantt Charts. These data points can then be used to calculate the utilisation of the equipment based on the actual processing time relative to the overall time to complete all batches.

Resource utilisation: Whether the resource is held in a pool or held as an item on a pallet, its utilisation can be monitored as with equipment. Alternatively Extend allows for the tracking of resource utilisation by connecting discrete even charts to the outputs of the resource blocks. The best option is to output the data to a software better suited to more comprehensive data analysis. Excel is perhaps the best candidate as most users will have a copy of the software readily available and data manipulation is relatively easy to carry out with intuitive chart outputs.

Ticket Generation: The desired output here is the number generated, the maximum open at any one time and the average open at any one time. A chart profiling ticket generation is also useful.

Lookahead

This is rather a complex piece of logic which is useful in representing manufacturing processes as it considers the trigger times for events. In the case of the BIOSYNT system it was initially thought necessary to create a lookahead logic capable of calculating trigger times for the preparation of intermediates and of equipment (egg. Equilibration in the case of chromatography units). A logic capable of considering any delays which may occur due to constraints such as labour shortage or equipment failure. However upon review of the series of constraints and rules under the model scope which determine the sequence of events, it was deemed that a look-ahead logic would be in effect redundant.

Shifts

There are two types of shift, that labour and operating. The labour shift can be modelled in two different ways based on the method of modelling the actual labour:
Method 1: if the labour has been modelled in a resource pool then the pool can be directly linked to a block capable of outputting a signal at particular points in time (in Extend this is the Value Schedule block which can in turn be linked to the database). This will control the availability of the labour based on system time or shift.
Method 2: if the labour has been modelled as an item then a gating system can be used to stop it from proceeding based on the shift times.
The operating shift can take many forms, for example the whole system will run on a particular shift pattern or individual equipment will have their own. Under both

circumstances, the two methods described for the labour shift can be considered, using the same selection criterion.

Data Transfer

Data generated by the model should be transferred to other software such as Excel for analysis. This is due to three reasons:

1) simulation modelling platforms do not offer the most intuitive methods of data analysis
2) charts and data stored in the model will add unnecessary complexity.
3) it is often best that the user not see the actual model, rather the inputs and outputs, which can be held elsewhere

In terms of data input, there must be sufficient complexity and data input to be relevant and useful to the user. It is only necessary to have user defined inputs where they will fall within the scope of the model. In other words the inputs required are those which allow the model to calculate/output parameters such as cost, time, yield/throughput and resource utilisation all of which have been listed under Model Logic. Input parameters should also be accessible without going deep into the model. There are two ways of achieving this

Method 1: input parameters can be entered into the database, either in Excel or Extend. The problem with the latter is that the user will be introduced to the underlying data source of the model.

Method 2: by creating a notebook level where all input locations are cloned from block dialog boxes. The user will only see the notebook containing a list of inputs and their meanings. The most important outputs can be cloned onto the notebook in the same way. This method is simple and user friendly, and does not require the opening of additional files such as the original database, thus making it the better option.

Database

The database will be used to hold all model parameters required for running. The best way to enter and manipulate the data is to first work in Excel. As a data handler the Excel software is much more equipped and user friendly. Furthermore it is more

or less universal software and therefore almost all users will have access to and knowledge of it (for purposes of future model changes). The data can then be imported into Extend and used for the running of the model. However difficulties arise when changes are made to the database in both Excel and Extend. It is important that the two be synchronised, or better still, for all changes to be made in Excel. Such problems can be altogether limited if all variable parameters (those likely to be changed at least) are entered into the notebook.

What's more, in order to achieve a greater level of standardisation, the database input can also be standardised whereby a template can be used each time. For example all manufacturing systems can use the same database template assuming that they all have activities, resources and entities. The database created for this model will most likely act as a template for future models.

USING THE STANDARDS STRUCTURES AND METHODS

The standards and methods proposed speed up the process of model development, making it easier to construct system representations by following the set guidelines.

The top level of the model was structured according to the standard in order for the system representation to be more intuitive to the user.

The next level is where the specific manufacturing process can be found. Figure B.1 shows the main activity blocks. Each contains the standard structure shown in Figure B.2, consisting of a set of sub-activity blocks which when linked to together form the overall main activity.

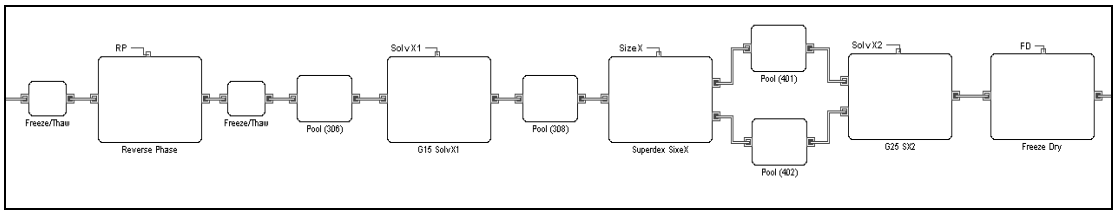


Figure B.1 Showing the BioSynT system product stream created by connecting together the main activity block

The order of activities runs from left to right and uses the IDEF0 convention of boxes and connectors. The connectors are labelled ItemIn and ItemOut rather than

BatchInand BatchOut as in manufacturing models the primary item may not be a batch (it may be a campaign or vial for example).

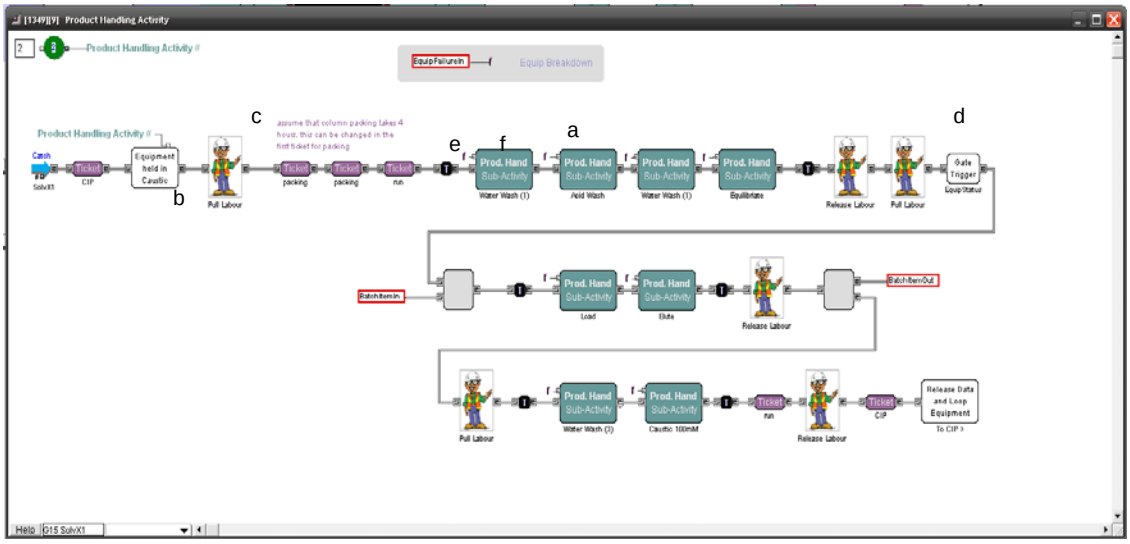


Figure B.2 Showing structure of a main product handling activity block. The blocks shown are (a) sub-activity block, (b) equipment hold, (c) labour pull, (d) labour release, (e) ticket tracker, (f) time stamp block.

The sub-activity blocks are quite generic and only adopt identity when a number is assigned corresponding to their position. For example, the first sub-activity will be given the identity number '1', the second '2' and so on. The database aware block then recognises the address of the parameter it must look up according to this identity.

There are two types of labour block, labour pull and labour release, which are generic in that the same block is used everywhere. The positioning of the labour blocks determines where the labour is pulled and where it is released back to the pool.

Once all activities have been completed the equipment is thrown to the external CIP block where it is cleaned before being sent back to be held in caustic.

In terms of pulling buffer material, for each step that requires a buffer, there is a 'buffer' block which uses flow to represent the filling activity of the material. For example, taking the same main activity block shown in Figure B.2, the equilibration

sub-activity requires the buffer Tris20. As Figure 6 shows, this buffer is pulled using a flow catch block and is received before the equilibration activity begins. The invisible link between the flow throw and catch blocks can be thought of as the piping which connects the buffer hold and column. Figure 7 shows the transfer out sub-activity block where the throw block connected to this catch is located. The Tris20 holding tank fills with the made-up buffer. When the column gets to its equilibration step, it requests the Tris20 which then flows to it. Once the amount required has been transferred, based on the column volume, flow is redirected to the waste tank to empty the remaining contents of the buffer hold tank. Using the flow method ensures that the holding tank is held up for the correct transfer period making the model far more representative of the real system making it more intuitive and relevant.

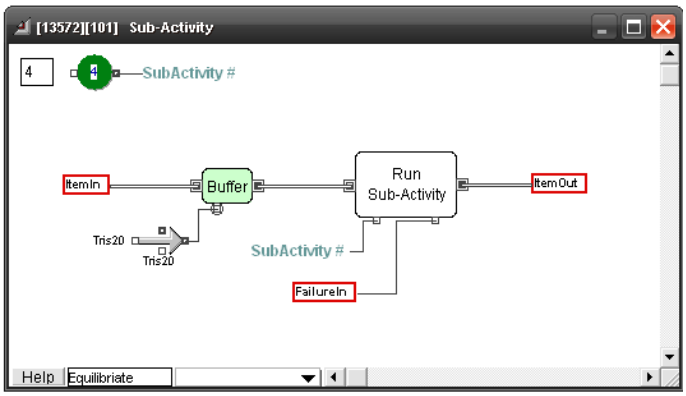


Figure B.3 Sub-Activity block with a 'Buffer' block which receives flow from the Tris20 holding tank

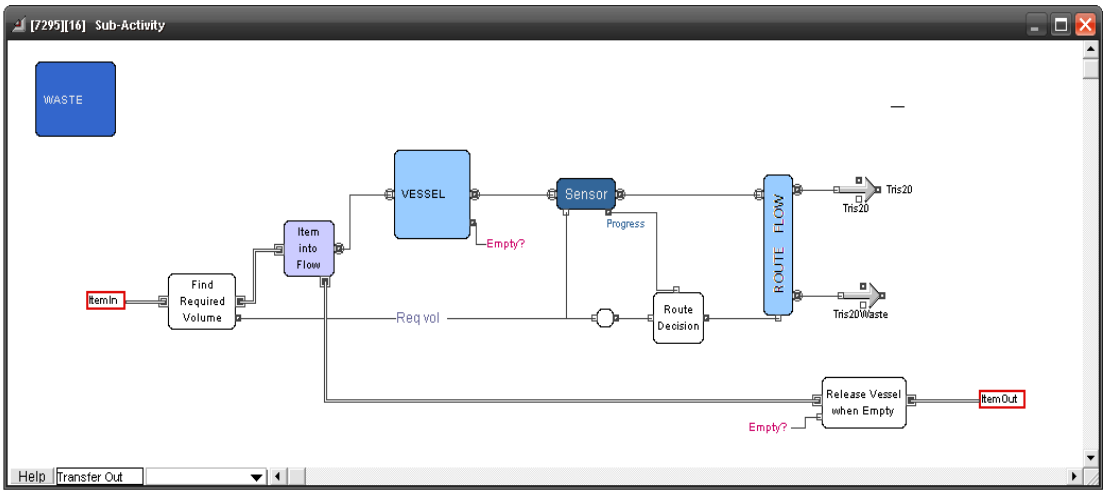


Figure B.4 Transfer-out Sub-Activity block where buffer is transferred from its holding tank to the receiving column.

The key inputs to the manufacturing model are categorised within the Excel database. These categories and data are summarised under 'inputs' in Figure B.5. The outputs of the model are also summarised. It should be noted that those shown here are specific to the scope of the capacity management issue addressed in this particular case study and will vary depending on the question being asked.

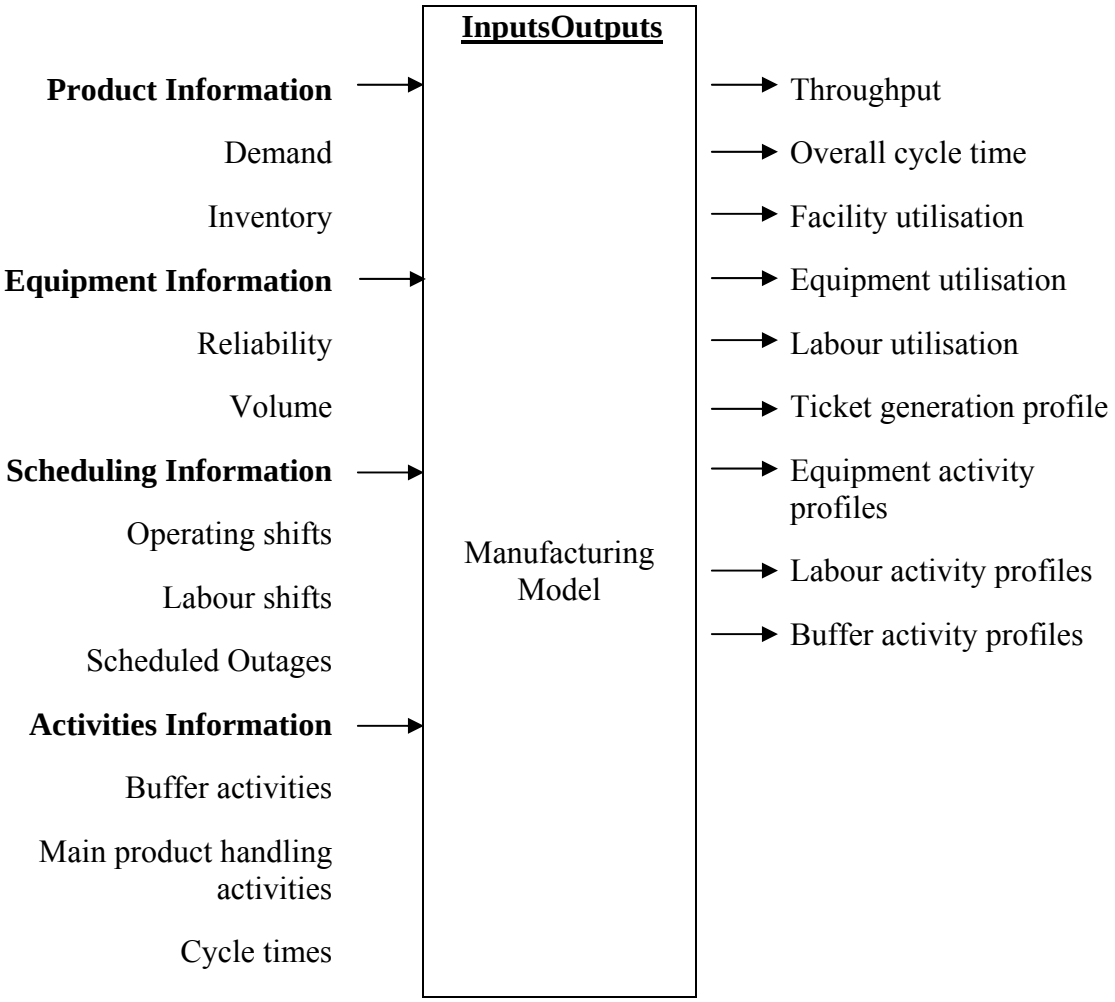


Figure B.5 Inputs to and outputs from the BioSynT model. The key inputs are product, equipment, scheduling, and activity information. The key performance measures are throughput, overall cycle time, resource utilisation and activity profiles

APPENDIX C

Data for Chapter 6

Four different scenarios were run for the case study presented in Chapter 6. These were:

1) Synchronised Turnaround procedure with 18 hour, 7 day operating shift
2) Synchronised Turnaround procedure with 24 hour, 7 day operating shift
3) Rolling Turnaround procedure with 18 hour, 7 day operating shift
4) Rolling Turnaround procedure with 24 hour, 7 day operating shift

The following tables show the results of those scenario runs

Appendix C.1 Results of different campaign schedules for synchronised turnaround 18hr7d shift

	Camp Seq 1			Camp Seq 2			Camp Seq 3			Camp Seq 4			Camp Seq 5			Camp Seq 6			
	A	B	C	A	C	B	B	A	C	B	C	A	C	A	B	C	B	A	
Day due	182	91	182	182	182	91	91	182	182	91	182	182	182	182	91	182	91	182	365
Completion day	301.7	360.9		300.7	358.1		74.5	352.9		74.63	132.7	364.8	55.69	334.2		55.69	133.4	364.6	364.6
Batch Demand in Q	20	5	2	20	2	5	5	20	2	5	2	20	2	20	5	2	5	20	33.27
Batches processed overall	40	2	0	40	2	0	5	40	0	5	2	32	2	40		2	5	32	25.27
Batches processed in Q	24	0	0	24.21	0	0	5	15.44	0	5	2	6.798	2	18.14	0	2	2.272	6.727	25.27
Days overdue	-31.15	269.9		-31.66	176.1		-16.5	31.71		-16.37	-49.31	66.75	-126.3	12.93	-91	-126.3	42.4	67	-0.4
CSL (Batches processed/demand)	1.2	0	-1	1.211	0	-1	1	0.772	-1	1	1	0.34	1	0.907	-1	1	0.454	0.336	0.76
Turnarounds	1			1			2			2			2			2			
Turnaround duration	8.67			8.67			17.3			17.3			17.3			17.3			
OEE Quality	100%			100%			100%			100%			100%			100%			
OEE Availability	100%			100%			100%			100%			100%			100%			
OEE Performance	89%			7%			26%			78%			30%			60%			
OEE	89.3%			7.0%			25.7%			77.9%			30.2%			59.6%			
Cleaning costs	260			260			520			520			520			520			
Start up cost for campaiging	120			120			120			240			120			360			
Total costs $	380			380			640			760			640			880			

	Camp Seq 7				Camp Seq 8				Camp Seq 9				Camp Seq 10				Camp Seq 11				Camp Seq 12			
	A	B	A	C	A	B	C	A	A	C	A	B	A	C	B	A	B	A	C	A	C	A	B	A
Day due	182	91	365	182	182	91	182	365	182	182	365	91	182	182	91	365	91	182	182	365	182	182	91	365
Completion day	159.9	237	361.1		159.8	236.7	294.5	363.5	161.1	218.8	364		159.8	218.8	294.4	363.1	74.5	236.5	294.1	363	55.69	217.9	294.8	363.2
Batch Demand in Q	20	5	20	2	20	5	2	20	20	2	20	5	20	2	5	20	5	20	2	20	2	20	5	20
Batches processed overall	20	5	15		20	5	2	4	20	2	17		20	2	5	4	5	20	2	4	2	20	5	4
Batches processed in Q	24	0	15	0	20	0	0	4	20	0.724	17	0	20	0.754	0	4	5	13.27	0	4	2	15.58	0	4
Days overdue	-22.15	146	-3.906	-182	-22.23	145.7	112.5	-1.544	-20.87	36.75	-0.958	-91	-22.23	36.75	203.4	-1.948	-16.5	54.5	112.1	-2.031	-126.3	35.86	203.8	-1.828
CSL (Batches processed/demand)	1.2	0	0.75	-1	1	0	0	0.2	1	0.362	0.85	-1	1	0.377	0	0.2	1	0.664	0	0.2	1	0.779	0	0.2
Turnarounds	2				3				2				3				3				3			
Turnaround duration	17.3				26.0				17.3				26.0				26.0				26.0			
OEE Quality	100%				100%				100%				100%				100%				100%			
OEE Availability	100%				100%				100%				100%				100%				100%			
OEE Performance	24%				30%				30%				39%				47%				49%			
OEE	23.7%				29.9%				30.2%				39.3%				46.5%				49.3%			
Cleaning costs	520				780				520				780				780				780			
Start up cost for campaiging	240				360				240				360				360				360			
Total costs $	760				1140				760				1140				1140				1140			

Appendix C.2 Results of different campaign schedules for synchronised turnaround 24hr7d shift

	Camp Seq 1			Camp Seq 2			Camp Seq 3			Camp Seq 4			Camp Seq 5			Camp Seq 6		
	A	B	C	A	C	B	B	A	C	B	C	A	C	A	B	C	B	A
Day due	182	91	182	182	182	91	91	182	182	91	182	182	182	182	91	182	91	182
Completion day	292	362.8		291.8	349.3		74.39	352.9		74.21	131.7	363.8	55.27	335.4		55.14	132	364.1
Batch Demand in Q	20	5	2	20	2	5	5	20	2	5	2	20	2	20	5	2	5	20
Batches processed overall	40	4	0	40	2	0	5	40	0	5	2	32	2	40	0	2	5	32
Batches processed in Q	24	0	0	24.95	0	0	5	15.46	0	5	2	6.934	2	18.1	0	2	2.332	6.892
Days overdue	-36.02	271.8	-182	-36.09	167.3	-91	-16.61	31.63	-182	-16.79	-50.29	65.75	-126.7	13.32	-91	-126.9	41.02	66.05
CSL (Batches processed/demand)	1.2	0	-1	1.247	0	-1	1	0.773	-1	1	1	0.347	1	0.905	-1	1	0.466	0.345
Turnarounds	2			2			2			2			2			2		
Turnaround duration	86.67			17.33			17.33			17.33			17.33			17.33		
OEE Quality	100%			100%			100%			100%			100%			100%		
OEE Availability	99%			100%			100%			100%			100%			100%		
OEE Performance	7%			8%			26%			78%			30%			60%		
OEE	6.6%			8.2%			25.7%			78.1%			30.1%			60.3%		
Cleaning costs	2600			520			520			520			520			520		
Start up cost for campaiging	120			120			120			240			120			240		
Total costs $	2720.1			640			640			760			640			760		

	Camp Seq 7				Camp Seq 8				Camp Seq 9				Camp Seq 10				Camp Seq 11				Camp Seq 12			
	A	B	A	C	A	B	C	A	A	C	A	B	A	C	B	A	B	A	C	A	C	A	B	A
Day due	182	91	365	182	182	91	182	365	182	182	365	91	182	182	91	365	91	182	182	365	182	182	91	365
Completion day	159.6	236.3	363.4		159.8	236.5	294	362.7	159.8	217.4	362.1		159.8	217.3	294.4	363.1	74.39	236.2	293.7	362.6	55.27	217	294.1	362.6
Batch Demand in Q	20	5	20	2	20	5	2	20	20	2	20	5	20	2	5	20	5	20	2	20	2	20	5	20
Batches processed overall	20	5	14		20	5	2	4	20	2	17		20	2	5	4	5	20	2	4	2	20	5	4
Batches processed in Q	24	0	14	0	20	0	0	4	20	0.771	17	0	20	0.773	0	4	5	13.3	0	4	2	15.67	0	4
Days overdue	-22.37	145.3	-1.644	-182	-22.23	145.5	112	-2.323	-22.17	35.38	-2.917	-91	-22.23	35.26	203.4	-1.948	-16.61	54.23	111.7	-2.418	-126.7	35.05	203.1	-2.37
CSL (Batches processed/demand)	1.2	0	0.7	-1	1	0	0	0.2	1	0.385	0.85	-1	1	0.387	0	0.2	1	0.665	0	0.2	1	0.783	0	0.2
Turnarounds	2				3				2				3				3				3			
Turnaround duration	17.3				26				17				26				26				26			
OEE Quality	100%				100%				100%				100%				100%				100%			
OEE Availability	100%				100%				100%				100%				100%				100%			
OEE Performance	23%				30%				31%				40%				47%				50%			
OEE	22.5%				29.9%				30.8%				39.6%				46.5%				49.4%			
Cleaning costs	520				780				520				780				780				780			
Start up cost for campaiging	240				360				240				360				360				360			
Total costs $	760				1140				760				1140				1140				1140			

Appendix C.3 Results of different campaign schedules for rolling turnaround 18hr7d shift

	Camp Seq 1			Camp Seq 2			Camp Seq 3			Camp Seq 4			Camp Seq 5			Camp Seq 6		
	A	B	C	A	C	B	B	A	C	B	C	A	C	A	B	C	B	A
Day due	182	91	182	182	182	91	91	182	182	91	182	182	182	182	91	182	91	182
Completion day	302.1	335.3	350.2	301.9	316.2	347.1	74.26	335.2	347.6	74.33	97.78	349.1	55.83	316.8	350.8	55.73	99.79	348
Batch Demand in Q	20	5	2	20	2	5	5	20	2	5	2	20	2	20	5	2	5	20
Batches processed overall	40	5	2	40	2	4	5	40	1	5	2	40	2	40	5	2	5	40
Batches processed in Q	24	0	0	24.12	0	0	5	16.52	0	5	2	13.41	2	19.34	0	2	4.003	13.25
Days overdue	-30.96	244.3	168.2	-31.07	134.2	256.1	-16.74	22.71	165.6	-16.67	-84.22	41.42	-126.2	4.292	259.8	-126.3	8.786	41.87
CSL (Batches processed/demand)	1.2	0	0	1.206	0	0	1	0.826	0	1	1	0.67	1	0.967	0	1	0.801	0.663
Turnarounds	2			2			2			2			2			2		
Turnaround duration	86.67			86.67			86.67			86.67			86.67			86.67		
OEE Quality	100%			100%			100%			100%			100%			100%		
OEE Availability	99%			99%			99%			99%			99%			99%		
OEE Performance	40%			40%			61%			89%			66%			82%		
OEE	39.6%			39.8%			60.3%			88.2%			65.0%			81.4%		
Cleaning costs	2600			2600			2600			2600			2600			2600		
Start up cost for campaiging	240			240			240			240			240			240		
Total costs $	2840.1			2840.1			2840.1			2840.1			2840.1			2840.1		

	Camp Seq 7				Camp Seq 8				Camp Seq 9				Camp Seq 10				Camp Seq 11				Camp Seq 12			
	A	B	A	C	A	B	C	A	A	C	A	B	A	C	B	A	B	A	C	A	C	A	B	A
Day due	182	91	365	182	182	91	182	365	182	182	365	91	182	182	91	365	91	182	182	365	182	182	91	365
Completion day	160	194.6	334.8	347.4	160.4	205.7	229.6	353	161.8	185.8	314.8	353.5	159.7	185.9	230.7	351.9	74.28	202.1	235.4	357.6	55.62	185.6	230.8	358.2
Batch Demand in Q	20	5	20	2	20	5	2	20	20	2	20	5	20	2	5	20	5	20	2	20	2	20	5	20
Batches processed overall	20	5	20	1	20	5	2	19	20	2	20	4	20	2	5	19	5	20	2	19	2	20	5	20
Batches processed in Q	24	0	20	0	20	0	0	19	20	1.686	20	0	20	1.702	0	19	5	16.85	0	19	2	19.44	0	20
Days overdue	-21.97	103.6	-30.2	165.4	-21.58	114.7	47.63	-11.97	-20.22	3.772	-50.15	262.5	-22.31	3.913	139.7	-13.14	-16.72	20.12	53.37	-7.433	-126.4	3.634	139.8	-6.771
CSL (Batches processed/demand)	1.2	0	1	0	1	0	0	0.95	1	0.843	1	0	1	0.851	0	0.95	1	0.843	0	0.95	1	0.972	0	1
Turnarounds	3				3				3				3				3				3			
Turnaround duration	123.8				118				141				131				135				140.8333333			
OEE Quality	100%				100%				100%				100%				100%				100%			
OEE Availability	99%				99%				99%				99%				99%				99%			
OEE Performance	55%				49%				71%				70%				70%				74%			
OEE	54.3%				48.1%				70.0%				69.0%				68.8%				73.2%			
Cleaning costs	3713				3540				4234				3935				4063				4225			
Start up cost for campaiging	360				360				360				360				360				360			
Total costs $	4073				3900				4594				4295				4423				4585			

Appendix C.4 Results of different campaign schedules for rolling turnaround 24hr7d shift

	Camp Seq 1			Camp Seq 2			Camp Seq 3			Camp Seq 4			Camp Seq 5			Camp Seq 6		
	A	B	C	A	C	B	B	A	C	B	C	A	C	A	B	C	B	A
Day due	182	91	182	182	182	91	91	182	182	91	182	182	182	182	91	182	91	182
Completion day	295.1	327.6	342.4	294	307.5	338.7	74.39	328.2	344.5	74.25	97.75	343.1	55.39	306	340.4	55.17	99.75	343.3
Batch Demand in Q	20	5	2	20	2	5	5	20	2	5	2	20	2	20	5	2	5	20
Batches processed overall	40	5	2	40	2	5	5	40	2	5	2	40	2	40	4	2	5	40
Batches processed in Q	24	0	0	24.76	0	0	5	16.96	0	5	2	13.74	2	20.21	0	2	4.018	13.51
Days overdue	-34.44	236.6	160.4	-34.99	125.5	247.7	-16.61	19.27	162.5	-16.75	-84.25	38.4	-126.6	-1.329	249.4	-126.8	8.755	39.52
CSL (Batches processed/demand)	1.2	0	0	1.238	0	0	1	0.848	0	1	1	0.687	1	1.011	0	1	0.804	0.675
Turnarounds	2			2			2			2			2			2		
Turnaround duration	86.67			86.67			86.67			86.67			86.67			86.67		
OEE Quality	100%			100%			100%			100%			100%			100%		
OEE Availability	99%			99%			99%			99%			99%			99%		
OEE Performance	40%			41%			62%			90%			67%			83%		
OEE	39.6%			40.9%			61.0%			88.7%			66.4%			81.9%		
Cleaning costs	2600			2600			2600			2600			2600			2600		
Start up cost for campaiging	240			240			240			240			240			240		
Total costs $	2840.1			2840.1			2840.1			2840.1			2840.1			2840.1		

	Camp Seq 7				Camp Seq 8				Camp Seq 9				Camp Seq 10				Camp Seq 11				Camp Seq 12			
	A	B	A	C	A	B	C	A	A	C	A	B	A	C	B	A	B	A	C	A	C	A	B	A
Day due	182	91	365	182	182	91	182	365	182	182	365	91	182	182	91	365	91	182	182	365	182	182	91	365
Completion day	159.4	198.2	327.9	354	159.5	204.7	235.2	361.9	159.7	185.9	315.3	359.9	159.5	186	229.9	358.6	74.53	190.5	216.6	359.5	55.36	184.3	229.4	351.3
Batch Demand in Q	20	5	20	2	20	5	2	20	20	2	20	5	20	2	5	20	5	20	2	20	2	20	5	20
Batches processed overall	20	5	20	2	20	5	2	20	20	2	20	5	20	2	5	20	5	20	2	20	2	20	5	19
Batches processed in Q	24	0	20	0	20	0	0	20	20	1.699	20	0	20	1.699	0	20	5	18.53	0	20	2	19.64	0	19
Days overdue	-22.62	107.2	-37.12	172	-22.48	113.7	53.15	-3.115	-22.28	3.95	-49.69	268.9	-22.48	3.99	138.9	-6.439	-16.47	8.526	34.62	-5.542	-126.6	2.303	138.4	-13.66
CSL (Batches processed/demand)	1.2	0	1	0	1	0	0	1	1	0.849	1	0	1	0.849	0	1	1	0.926	0	1	1	0.982	0	0.95
Turnarounds	4				3				3				3				3				3			
Turnaround duration	167.6				134.8				143				131				124				140.4419824			
OEE Quality	100%				100%				100%				100%				100%				100%			
OEE Availability	98%				99%				98%				99%				99%				99%			
OEE Performance	55%				50%				71%				71%				73%				73%			
OEE	54.0%				49.3%				70.2%				70.2%				72.2%				72.2%			
Cleaning costs	5029				4042.634929				4291				3944				3733				4213.259472			
Start up cost for campaiging	360				360				360				360				360				360			
Total costs $	5389				4402.634929				4651				4304				4093				4573.259472			