



## Modelling biopharmaceutical manufacture: Design and implementation of SIMBIOPHARMA

Suzanne S. Farid<sup>a,\*</sup>, John Washbrook<sup>b</sup>, Nigel J. Titchener-Hooker<sup>a</sup>

<sup>a</sup> *The Advanced Centre for Biochemical Engineering, Department of Biochemical Engineering, University College London, Torrington Place, London WC1E 7JE, UK*

<sup>b</sup> *Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK*

Received 7 December 2004; received in revised form 21 July 2006; accepted 12 October 2006

### Abstract

This paper presents the implementation of a conceptual framework, for modelling a biopharmaceutical manufacturing plant, into a prototype decision-support tool, SIMBIOPHARMA. The tool's scope covers the ability to evaluate manufacturing alternatives in terms of cost, time, yield, resource utilisation and risk. Incorporating uncertainty means that investment appraisal can be based on both the expected outputs and the likelihood of achieving them. A hierarchical approach to represent the key activities in a manufacturing process is introduced. Emphasis is placed on how a closer integration of bioprocess and business process modelling can be achieved by capturing common information in an object-oriented environment, G2 (Gensym Corporation, Cambridge, MA). The key features of SIMBIOPHARMA are highlighted; these include interactive graphics, task-oriented representation and dynamic simulation which create a much more flexible environment for modelling processes. Examples of typical outputs generated by SIMBIOPHARMA, when addressing the impact of manufacturing options on strategic operational and financial indicators, are given. © 2006 Elsevier Ltd. All rights reserved.

**Keywords:** Object-oriented; Decision-support tool; Computer-aided simulation; Biopharmaceutical batch manufacture; Costs; Uncertainty

### 1. Introduction

As the number of protein therapeutics in clinical evaluation continues to rise, more efficient manufacturing strategies are being sought to cope with the increasing demands (Pisano, & Wheelwright, 1995; Byrom, 2000; Kelley, 2001; Chadd & Chamow, 2001). However, selecting an effective manufacturing strategy requires exploring the balance between striving for improved operational efficiencies and the subsequent implications on cost, time and risk. Unless the technical and financial aspects are addressed explicitly, a company risks making decisions that are not best-suited to achieve all goals. Simulation tools can be used to enhance the understanding of manufacturing systems and act as a test bed for their evaluation. The design of such a tool provided the motivation for this research.

Historically, attractive returns have made companies emphasise speed to market rather than focus on improving process

economics (Sadana, & Beelaram, 1994). The financial squeeze on biopharmaceutical companies has triggered renewed interest in manufacturing and its contribution to business performance. Hence, emphasis on estimating and improving manufacturing costs is growing. However, at present no software package allows both process modelling and effective decision-support that relates bioprocess decisions to all the following strategic business issues: costing, resource management and risk.

In this paper, a conceptual framework to facilitate the evaluation of manufacturing alternatives, and its implementation into a software tool, designated SIMBIOPHARMA, are presented. The proposed framework aims to bridge the gap between the process and business perspectives of manufacturing, as well as incorporating risk analysis. For the implementation into the decision-making software tool, emphasis is placed on how a closer integration of bioprocess and business process modelling can be achieved by capturing common information in an object-oriented environment. The aim is to provide a platform for rapid evaluation of biopharmaceutical manufacturing alternatives in terms of cost, time, yield, resource utilisation and risk. Consequently the trade-off between financial and operational

\* Corresponding author. Tel.: +44 20 7679 4415; fax: +44 20 7209 0703.  
E-mail address: [s.farid@ucl.ac.uk](mailto:s.farid@ucl.ac.uk) (S.S. Farid).

parameters can be examined explicitly, the ramifications of resource delays captured and the impact of uncertainties on the robustness of the solutions can be ascertained.

## 2. Domain description

Prior to commercial manufacture, the production of a biopharmaceutical drug candidate will be required at several stages during its development cycle. Manufacturing issues must therefore be addressed early in the development process to ensure sufficient supplies are produced to the correct standard, enabling deadlines for clinical trials and product launch to be met (Clemento, 1999; Byrom, 2000; Savage, 2000). Bulk drug manufacture must also support additional critical steps, such as process and assay development, stability programs, pre-clinical testing and registration samples (Clemento, 1999; Bernstein & Hamrell, 2000). Manufacturing decisions are further complicated by the need to comply with stringent current good manufacturing practices (cGMP), as well as the unique process validation requirements of biotech facilities. These regulations are critical for controlling the quality and safety of biologics. Although manufacturers of all drugs must adhere to cGMPs, biologic drugs suffer more serious consequences if failures in engineering and procedural controls occur (Copmann et al., 2001). These can lead to contamination or product alteration, compromising product safety for clinical use. Extensive documentation must be provided to comply with these requirements (Gregersen, 1995).

Manufacturing of biopharmaceutical products is primarily by batch processes. As an illustration, one might consider the manufacture of biopharmaceuticals derived from mammalian-based processes. Here each batch is produced by a series of operations that proceed from fermentation (cell culture) through to product recovery (e.g. centrifugation, microfiltration) and finally purification (e.g. chromatography, ultrafiltration), in a train of vessels. Additional manufacturing operations are involved indirectly in the production of a batch. These include the preparation of intermediate materials, such as fermentation media and buffer solutions. Equipment used to produce a batch also need to be prepared; examples include cleaning-in-place (CIP) and sterilising-in-place (SIP) operations, as well as chromatography column preparation activities. Further support functions include the essential quality control activities, such as product assays, equipment and process validation, and the documentation required to complete each batch record for regulatory purposes.

Successful operation of a production facility requires both a technical understanding of the process and a command of the logistics of operations affecting the business. On the process front, improvements in technical performance, such as higher product yields, can be achieved by adapting the sequence of operations, the equipment specifications and operating conditions and examining the effects on the properties and composition of the process and the batch cycle time. On the business front, enhancements in the management of operations require an analysis of the scheduling of operations, the utilisation of resources (such as personnel, equipment, materials and utilities), the overall productivity, the cost of goods and profitability. In addition,

there are uncertainties in biopharmaceutical manufacture that impact on product throughput, costs and delivery time. These are associated with factors, such as yields, processing times, resource costs, product demands, contamination rates and success in clinical trials. A computer-aided tool capable of capturing and integrating these process and business needs, as well as incorporating risks, can help provide a rational basis for confident decision-making in manufacturing.

## 3. Modelling biopharmaceutical manufacture

In the past, process modelling (e.g. mass balances, plant design, capacity management) has usually been treated separately from the modelling of business issues (e.g. investment decisions, production planning, risk assessment). A brief overview of process and business simulation efforts for bioprocesses is described below.

### 3.1. Process simulation

Computer-aided process simulation has become a standard tool to plan, design, optimise and evaluate chemical processes (Gritsis & Titchener-Hooker, 1989). However, the application of computer simulation has been slower in the biopharmaceutical industry and the reasons for this have been highlighted by several authors (Gritsis, & Titchener-Hooker, 1989; Bogle et al., 1996; Petrides, Calandranis, & Cooney, 1995; Polakovic & Mandenius, 1996; Gosling, 1996). Despite these problems, the potential benefits of computer-aided process design tools have been gaining increasing recognition in the bioprocessing industry which is confronted with limited finances and pressing timelines. It is now widely acknowledged that such tools can be used to help reduce process development times by allowing different process sequences and operating conditions to be examined inexpensively via modelling, thus saving time and reducing costly pilot-scale trials (Gritsis, & Titchener-Hooker, 1989; Petrides, & Sapidou, 1997). The advantage of such a computer-based tool is that it permits bioprocess decisions to be better placed in the context of business processes and this feature is gaining increased recognition. The remainder of this section provides examples of the approaches adopted to address these two types of modelling in the bioprocessing industries.

The development of computer-aided design tools for bioprocessing began in the mid 1980s (Petrides et al., 1995). Published work in this area indicates that two main routes for developing bioprocess simulators have been pursued. The first has concentrated on adapting existing process simulators from the chemical sector (e.g. AspenTech's BPS). The second has developed detailed models of individual unit operations verified using experimental data (e.g. Gritsis, & Titchener-Hooker, 1989; Zhou, Holwill, & Titchener-Hooker, 1997; Varga, Titchener-Hooker, & Dunnill, 2001). In addition there has been work specifically targeted at batch processes but primarily developed for chemical process development (e.g. Stephanopoulos, Ali, Linninger, & Salomone, 2000). However, little work has been presented that allows modelling of complete bioprocesses whilst encompassing the process performance as well as the accom-

panying capital investment, operating costs, resource utilisation and uncertainties. This paper investigates the possibility of combining process, cost and risk modelling in a single framework. This should permit investment appraisal to be based on both financial and operational metrics as well as the likelihood of achieving certain threshold values.

### 3.2. Cost modelling

The biochemical engineering textbook by Atkinson and Mavituna (1991) provides complete examples of how to estimate the production costs for different traditional processes, including intracellular enzyme and penicillin production. Datar, Cartwright, and Rosen (1993) illustrated how the expression system could have a major impact on the total number of required processing steps and hence the economic viability of a product. They compared the costs of producing the recombinant protein, rTPA, in *E. coli* and CHO cells. Detailed attempts at deriving costs for transgenic expression technologies are also starting to appear in the literature, anticipating ton-scale production of antibodies (e.g. Mison, & Curling, 2000). The remaining contributions in the literature tend to focus primarily on the cost of chromatographic separations rather than whole processes (e.g. Sofer, & Hagel, 1997). In addition to assessing the production costs of biopharmaceuticals, certain publications have also assessed profitability using standard discounted cash flow techniques (e.g. Novais, Titchener-Hooker, & Hoare, 2001).

### 3.3. Risk modelling

Traditional process modelling and investment analysis techniques assume all outputs occur with certainty and hence they are 'no-risk' performance measures. However, manufacturing decisions are often made in an uncertain environment characterised by technical and market-related risks. For example, common uncertainties in manufacturing systems include yields, processing times, set-up times and batch failures (Banks, 1998), as well as the amount of investment required, the cost of inputs into production and product demands. Formal methods of incorporating such risks require a subjective assessment of the probability distributions of all the key variables, typically based on historical data or through soliciting advice from industrial experts. Two of the methods for using this information are 'risk adjustment' (Moilanen, & Martin, 1996) and 'Monte Carlo simulation', which has been used increasingly in various business situations (Hayes, & Wheelwright, 1984).

## 4. Scope of framework

Defining the scope of the modelling effort was a key initial stage required to focus simulation efforts and to ensure the breadth of the analysis was not too wide that it became too complex to handle. The scope of the modelling framework was defined as follows:

- To prototype manufacturing operations in a multi-product biopharmaceutical facility, operating on a campaign basis,

- To investigate supply-and-demand issues of drugs for clinical trials or commercial manufacture,
- To evaluate manufacturing alternatives in terms of cost, time, yield, resource utilisation and risk.

Manufacturing alternatives that the simulation model should be able to address were:

- Facility decisions – e.g. the use of a stainless steel versus a disposables-based facility,
- Process decisions – e.g. the use of expanded bed chromatography versus conventional product recovery techniques,
- Capacity decisions – e.g. selecting the optimum downstream capacity for a potentially successful new drug.

## 5. Modelling approach

### 5.1. Hierarchical framework

The framework used a hierarchical approach to represent the key activities in a manufacturing process through a series of levels. Such approaches have often been applied to systems in other sectors (Manivannan, 1998; Lakshmanan, & Stephanopoulos, 1998; Subrahmanyam, Pekny, & Reklaitis, 1996; Puigjaner, & Espuna, 1998; Johnsson, & Arzen, 1998). More recently hierarchical decompositions have also been employed by researchers to represent all the phases of drug development (Karri, Davies, Titchener-Hooker, & Washbrook 2001; Rajapakse, Titchener-Hooker, & Farid, 2005). The hierarchical framework presented in this paper focuses specifically on the manufacturing operations.

The hierarchical framework was modular and extensible allowing further levels of detail to be added as required. Hence, the procedure to manufacture a product could be represented at different abstraction levels according to the desired goals of the user. Modelling high-level activities helped provide an overview of the process and a summary of the key operational and financial parameters. With more information, each high-level activity could be broken down into sub-tasks that generated more accurate estimates of the key parameters. The use of sub-tasks also permitted parameters, not considered at higher levels (e.g. steam usage), to be modelled. This hierarchical approach to modelling enabled selected activities to be examined in great detail. This was a computationally efficient approach and also reflected the fact that process data is often sparse. An example of the hierarchical levels used to describe product manufacture is illustrated in Fig. 1.

Each manufacturing campaign was composed of a set of batches to meet the product demand. As indicated in Fig. 1, each batch was defined by a set of recipes, which interact with each other to produce a single batch. Each recipe comprised a procedure and its resource requirements. Each recipe procedure could then be specified, in a hierarchical structure, in terms of its unit procedures, and/or operations, and/or phases. Hence, a model could be considered complete at any level of the hierarchical framework; the level of detail adopted is dictated by the purpose of the simulation and the information available.

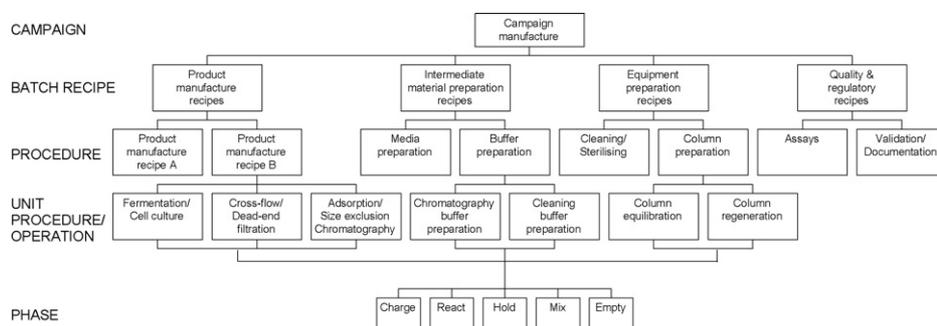


Fig. 1. Hierarchical representation of manufacturing tasks in a biopharmaceutical plant. Each manufacturing campaign is composed of a set of recipes, which interact with each other to produce a single batch. The level of detail adopted is dictated by the purpose of the simulation and the information available.

A closer examination of the hierarchical framework proposed specifically for biopharmaceutical manufacture is given below. The domain description highlighted the multiple types of activities required to manufacture a product. Performing all these operations within a single recipe can be inefficient and complicated to modify. The hierarchy was therefore configured to use separate recipes to manufacture the product, prepare the equipment, prepare the intermediate materials and to perform support quality control activities. This is indicated in the “batch-recipe” level in Fig. 1. This provides greater flexibility than grouping all activities together within a single recipe considering that often equipment, media and buffer are prepared prior to, or concurrent with, the activities that directly handle the batch of final product.

The ancillary recipes could be scheduled for use by the main product manufacture recipe. For example, the product manufacture recipe could acquire equipment for processing use and release it after use for cleaning by another recipe. This permits dirty equipment to be cleaned by a CIP recipe which can be applied as often as is necessary. Another advantage of the approach is that certain ancillary activities, such as the preparation of intermediate materials, are not always dedicated to a specific campaign. Similar separation of product recipes from recipes performing ancillary tasks can be found in Bastiaan (1998) and Cowl (1997).

As illustrated in Fig. 1, each recipe procedure could be modelled in greater detail in terms of its constituent unit procedures. For example, the product manufacture recipe procedure could comprise unit procedures, such as “fermentation”, “filtration” and “chromatography”. Similarly the equipment preparation recipes could be modelled as unit procedures that are specific to certain equipment, e.g. “CIP fermenter”. Under certain circumstances it may be desirable to define each unit procedure more precisely in terms of its individual operations. For example, the sequence of product-handling operations in a fermentation task could be “agitate”, “heat”, “react”, “feed addition” and “discharge”.

All the hierarchical views of the activities in a manufacturing facility needed to be reconciled with the top-level view. This was made possible through the use of common parameters running through all the levels, namely cost, time and product mass. However, lower-level views also provided valuable outputs and were therefore of use in their own right, e.g. the unit operation

level was used to obtain resource utilisation profiles. A summary of the typical outputs from each level is shown in Table 1.

### 5.2. Key features

To satisfy both process and business applications each level specified the manufacturing process in terms of the manufacturing tasks, the resources within the plant, and the process streams that flow between material-handling tasks. Outside these core knowledge requirements existed characteristics that were more specific to each individual application. For process applications, examples include mass balance data and associated mathematical procedures. Business applications require cost data and knowledge of resource availability and utilisation. The developed framework permits the user to investigate different production strategies in terms of the process efficiency, mass and project throughput, resource demands and bottlenecks, and the cost of goods.

### 5.3. Key parameters

All the features described in the previous section had input and output parameters, or attributes. The following measures were considered critical for assessing the capabilities and limitations of a manufacturing strategy: cost, time product mass

Table 1  
Summary of typical outputs from each level

Level	Outputs
Campaign	Total number of campaigns Total cost of goods Total fixed capital investment Summary of recipe direct costs Cumulative duration of campaigns Cumulative product mass supplied
Batch recipe procedure	Total number of batches per campaign Recipe direct cost Recipe duration Product mass supplied per batch & campaign
Unit procedure/ operation	Task direct cost Task duration Process stream compositions Resource utilisation profiles

and risk. These are closely interrelated; the ideal manufacturing plant should operate within a timeframe that maximises the mass throughput rate while being cost-effective and with acceptable risk.

#### 5.4. Stochastic simulation

Integral to this study has been the incorporation of risk and uncertainty so that investment appraisal could be based on both the expected outputs and the likelihood of achieving certain threshold values. Relying solely on deterministic results could lead to the risk of incorrect conclusions that could be damaging to a company's overall economic performance. To measure the risk of a proposed investment, probability distributions could be assigned to all the key uncertain variables. While historical data could be used to identify suitable distributions, most uncertainties were subjective estimates from informed experts. The Monte Carlo simulation technique would then be used to determine the resulting frequency distributions of the output summary measures. Consequently several possible outcomes are simulated and their impact on the outputs measured as well as their likelihood of occurrence. This facilitates choosing the most robust manufacturing option and allows questions, such as "what is the chance of exceeding the cost budget?" and "what is the probability of failing to meet our desired plant output?" to be answered. Hence, incorporating uncertainties also provides a more realistic ranking of manufacturing alternatives.

### 6. Process and cost models

#### 6.1. Process models

The process models were selected primarily to describe a manufacturing process based on mammalian cell culture. Care was taken to construct or select models that described outputs that facilitated rapid assessment of manufacturing alternatives, without requiring data inputs that were complex or that necessitated specially designed experiments for their estimation. This was key to enable facility, process or capacity decisions that considered both the technical performance and the implications on resources and costs to be made at early design stages.

The process models comprised a combination of design equations, which reflected design constraints arising from the process and equipment, and mass balances taken over material-handling tasks. The basis of each process model is summarised in Table 2. Examples of the equations are given in Appendix A.

#### 6.2. Cost models

Fixed capital investment and the cost of goods were used to compare different production strategies, are the. Simple cost models were initially proposed to provide rapid estimates of these metrics. The open architecture of the framework permits more detailed and accurate models to be added to complement these as and when additional data becomes available from the biopharmaceutical industry.

##### 6.2.1. Fixed capital investment

The fixed capital investment was approximated using the Lang factorial method (Lang, 1948). For chemical engineering facilities, values in the range of 3–5 are often recommended (Sinnott, 1993; Peters, & Timmerhaus, 1991). Examples of capital cost breakdowns in the literature (Osborne, 1997; Hamers, 1993; Nelson, 1998) have suggested factors in the range of 6–7 as more typical for biopharmaceutical plants. In addition, discussions with industrial experts have indicated that factors in the range of 4–7 are more suitable for the biopharmaceutical sector (A. Sinclair, BioPharm Services, Bucks, England, personal communication). These provide default input values for the model.

##### 6.2.2. Cost of goods (COG)

The cost of goods comprised both the direct costs accrued based on resource utilisation and indirect manufacturing costs, e.g. maintenance, that are incorporated as investment-related costs. The duration of campaigns and the mass throughput also influenced the operating costs. The tool employed the model indicated in Table 3 to calculate the cost of goods (COG). The COG model was adapted to reflect the extra costs associated with current good manufacturing practices (cGMP) for biopharmaceutical plants. The cost of goods was defined as the manufacturing cost, thus excluding general expenses, such as

Table 2  
 Basis of process models

Unit operation	Basis – simple models	Key outputs
Fermentation	Stoichiometry; Extent of reaction	Broth composition
Centrifugation	Solids carry-over; Solids volume-fraction in sediment	Sediment composition; Supernatant composition
Micro-ultra-filtration	Flux; Rejection coefficients	Permeate composition; Retentate composition; Membrane area <i>or</i> Concentration factor <i>or</i> Processing time
Diafiltration	Flux; Rejection coefficients; Number of diafiltration volumes <i>or</i> Contaminant removal-fraction	Permeate composition; Retentate composition; Membrane area <i>or</i> Concentration factor <i>or</i> Processing time; Number of diafiltration volumes <i>or</i> Contaminant removal-fraction
Dead-end filtration	Flux; Rejected particle-fraction; Particle volume-fraction in retentate	Permeate composition; Retentate composition; Membrane area <i>or</i> Processing time
Chromatography	Flowrates; Yields	Product stream composition; Waste stream composition; Processing time; Buffer volumes

Table 3  
Cost of goods model

Cost category	Value	
Direct	Direct raw materials	f (utilisation)
	Miscellaneous materials	$0.5 \times$ Direct raw materials
	Direct utilities	f (utilisation)
	Operating labour	f (utilisation)
	Supervisors	$0.2 \times$ Operating labour
	QCQA	$1.0 \times$ Operating labour
	General management	$1.0 \times$ Operating labour
Indirect	Maintenance	$0.1 \times$ Capital investment $\times$ Project duration
	Local taxes	$0.02 \times$ Capital investment $\times$ Project duration
	Insurance	$0.01 \times$ Capital investment $\times$ Project duration
	Depreciation	Capital investment/Depreciation period $\times$ Project duration
	General utilities	Cost per unit area per year $\times$ Facility size $\times$ Project duration
Total	Cost of goods	

\*QCQA = Quality Control & Quality Assurance.

sales and administration costs. The direct (variable) costs were computed based on the utilisation of the material, utilities and staff resources. The indirect costs (fixed overheads) were mostly derived from the capital investment. Since the tool was intended for evaluating the effects of different production strategies it was found to be more useful to base the staff costs on their utilisation rather than considering them as a fixed annual salary-based cost. Monitoring the direct utilities cost does not account for the ongoing utility charges for running the manufacturing facility. For example, HVAC systems are critical to controlling airborne particulate levels and pressure differentials in different rooms so as to prevent contamination and securely contain micro-organisms used for production. An extra cost category termed “general utilities” was added to account for this. Advice on a suitable correlation for this category was sought from industrial experts (A. Sinclair, Biopharm Services, Bucks, England; M. Sawyer, Lonza Biologics, Portsmouth, NH, personal communication). The cost was derived as a function of the facility size or floor area. The fixed overhead costs were allocated to a project based on its occupancy time in the manufacturing plant.

Apart from the direct resource costs, the remaining cost categories were each estimated as a function of other costs. The default values for the factors used in these functions are indicated in Table 3; they provide a rapid estimate of the distribution of costs in a biopharmaceutical plant. These values were derived from literature or expert sources as indicated below. Pugh (1998) provides estimates of the miscellaneous (indirect) material costs (e.g. safety clothing) as 33–67% of the direct raw material costs for cell culture facilities. A value of 50% was taken as the initial default value. The staff for supervision, quality and general management were estimated based on the operating labour cost as suggested in chemical engineering textbooks (Peters, & Timmerhaus, 1991). However, the actual value of the fac-

tor for quality staff was increased to reflect the higher burden on quality control activities in the biopharmaceutical industry. The new value for the factor was derived after soliciting advice from industrial experts. Some (C. van Hoorn, Merck & Co., West Point, PA; N. Henschel, Boehringer Ingelheim Pharma KG, Germany, personal communication) suggested that relative to a chemical pharmaceutical facility, the quality assurance demands would be similar in a biopharmaceutical plant, whereas the quality control staff would be two to three times greater. Others have suggested that for every operator, there should be two quality control staff. In chemical engineering textbooks, quality costs are estimated as one-fifth of the operating labour cost. Using the advice from industrial experts suggests that a factor of 0.4–2 would be more appropriate. A default value of 1 was selected. When quality operations are modelled explicitly this factor can be reduced or removed to avoid duplications of costs. The default value for the annual cost of general utilities per unit floor area was assumed to be \$300/m<sup>2</sup> (M. Sawyer, Lonza Biologics, Portsmouth, NH, personal communication). The factors for the remaining fixed overheads, such as maintenance were assumed to be similar to those for process engineering facilities.

In addition to calculating the cost of goods on a cost category basis, the framework also permitted the calculation of the direct cost of goods on a task basis, as mentioned earlier. Viewing costs on a unit operation basis can be more useful to a process development team since it highlights where to focus cost reduction efforts.

## 7. Data collection

As highlighted in the previous section, parameters in the models within the framework have default values obtained from a range of sources: industrial experts, historical data, literature and vendors. Data collection involved a series of discussions with experts (J. Birch, D. Sherwood, J. Bonnerjea at Lonza Biologics, Slough, England). In addition, these experts were consulted so as to validate the structural model assumptions through discussions of the manufacturing operations in biopharmaceutical plants. The experts were also called upon to validate whether the results of the simulation model were reasonable. Default values for parameters in the process and cost models, such as linear velocities in a chromatography step and cost factors, were built into the models with the flexibility to modify them.

## 8. Selecting a software platform for implementation

It was necessary to summarise the required capabilities of the software platform(s) and assess which of several platforms would be best-suited to implement both the logistical models, describing the interactions between tasks and resources over time, and the mathematical process models, used to determine the composition of the process streams.

### 8.1. Requirements specification

The conceptual framework identified the key features, e.g. tasks, and parameters, such as cost required to represent the

Table 4  
Requirements specification for the tool

Requirement type	Specification	
Representation of declarative and procedural knowledge	Tasks and their characteristics	
	Resources and their characteristics	
	Material flow and its characteristics	
	Relationships between tasks, resources and material flow	
	Sequences of tasks	
	Resource requirements for each task	
	Calculation procedures for mass balances and costing	
	Variables for the calculation procedures	
	Time	
	Hierarchical views of the tasks	
	Risk/uncertainty: stochastic variables defined using probability distributions	
	Dynamic simulation	Dynamic simulation of task sequences
		Dynamic allocation of resources to tasks
Dynamic invocation of calculation procedures to compute compositions and costs		
Dynamic invocation of procedures to compute resource utilisation statistics		
Monte Carlo simulation		
Single-threaded, multi-threaded and parallel processing		
Flexible development environment	Graphical user-interface	
	Modular Extensible	

process and business perspectives of manufacturing processes of biopharmaceuticals. This permitted the development and refinement of a “requirements specification” that describes what the software platform should be able to accomplish. A summary of the requirements specification is shown in Table 4.

The possibility of using SuperPro Designer (Intelligen, Scotch Plains, NJ), a commercially available bioprocess simulation package was investigated. Key advantages were found to be that it was relatively simple and fast to set up a flowsheet for an entire process. It also had default values for many of the input data required for simulation that could be used when experimental data were not readily available or as a reference to check against. However, one of the key disadvantages was that it could not capture the desired dynamic behaviour capabilities related to time-dependent operations (Table 4). An example of information that needs to be updated in “real-time” is resource availability, since this affects the allocation of resources to tasks over time. This is important for tasks competing for specific resources since the tasks can become delayed by the non-availability of a resource. In addition, the availability of equipment resources is further constrained by their status which can change between “clean”, “sterile” and “dirty”. SuperPro Designer did not account for the impact of delays due to resource constraints and hence had limited logistical capabilities. These conclusions are echoed by Shanklin, Roper, Yegneswaran, and Marten (2001) in their evaluation of Aspen Batch Plus and SuperPro Designer for mod-

elling a vaccine manufacturing process; they report that neither package accounts for constraints, such as available labour or time limitations between cleaning and processing. Another key limitation for the purposes of this work was the inability to incorporate probability distributions to represent the uncertainty in parameter values. Furthermore, it only offered pre-specified functionality with no option to create user-defined models. This meant that it was not possible to model newer unit operations, such as expanded bed chromatography, or to customise existing models. On the other hand, object-oriented programming languages were found to have features that support all these required capabilities. Similar conclusions were related by Knutilla et al. (1998), when examining several platforms for specifying manufacturing processes.

## 8.2. Implementation strategy using an object-oriented approach

Gensym’s G2<sup>®</sup> (Gensym Corporation, Cambridge, MA) was selected as the most suitable environment in which to build the models describing the process and business perspectives of biopharmaceutical manufacture.

G2 is a graphical simulation tool that enables modelling of a number of key manufacturing and business features, such as resource utilisation and costs. Both types of applications require common information, such as the process sequence, the unit operation models, the design data, the relationships between objects and the constraints on the process due to time, cost and resource availability. This graphically- and object-oriented knowledge-based tool can be used to capture and formalise all these common properties. A particular advantage of such an object-oriented language is its rapid prototyping capability, which facilitates the development of applications in a modular fashion, where modules are reusable. A benefit of the knowledge-based approach is that more of the information is in an explicit, declarative form, accessible by both developers and end-users, rather than buried in procedural code (Stanley, 1994). This facilitates error detection. A further and powerful property of such knowledge-based systems is the inheritance in a class hierarchy, which simplifies development and maintenance by abstracting common properties and behaviours.

ReThink<sup>®</sup> (Gensym Corporation, Cambridge, MA), an application that runs in G2’s object-oriented programming environment, contains useful pre-built features, such as basic tasks and resources, to enable business process re-engineering. New features pertinent to biopharmaceutical manufacture can be added by extending and customising the building blocks in ReThink, using the G2 programming environment. Theoretically then when modelling a manufacturing process at the unit procedure/operation level, ReThink can be configured to allow users to build simulation cases that comprise the production process sequences as interconnected tasks, each with a duration and cost and each related to the resources it requires. This would monitor the cost and time outputs. The advantage of this task-oriented approach is that it focuses more on the logistics of production unlike the unit-oriented approach that focuses more on the plant schematic and promotes a clear linkage between the

manufacturing activities, their resource requirements and costs incurred through the use of the allocated resources. This can provide a better understanding of where costs originate and how improvements can be made to the process. Another benefit of the task-oriented approach is that it facilitates a hierarchical analysis of the process by describing the manufacturing activities at various levels of abstraction. Resource requirements are expressed explicitly for each task providing a clear visualisation of the tasks and their required resources. Such an approach confers a higher level of flexibility than implicit resource representation since the resource requirements of a task are not fixed in its properties but can be set at any time without disrupting the tool infrastructure.

The technical performance of each task and hence the material flow to each task was dictated by mathematical models for the unit operations. The values of these performance variables could either be determined by specifying the mathematical models in G2 or by data-seeking from an external simulator containing the unit operation models. A key implementation decision was to determine the optimum method of incorporating the output of the mathematical process models within this decision-support software. Since the mathematical complexity of the simple process models is within G2's capabilities, the process and logistical models were created all in G2; this avoided the problems of interfacing between different packages and offered the advantages of an open architecture for future additions of features. The tool was designed and implemented in ReThink.

## 9. Tool overview

In designing SIMBIOPHARMA the challenge was to represent the declarative and procedural knowledge required in a sufficiently robust manner, so as to enhance the efficiency, maintainability and reusability of the application. It was necessary to customise ReThink to model the processes of biopharmaceutical manufacture using the G2 language. The components of the tool built with this language possess the benefits of object-oriented design, including the use of encapsulation, inheritance and polymorphism.

The mapping of the key features of the framework into object classes was facilitated by the principles of object-oriented design. The development of the class hierarchies was more involved. It was guided by the requirements of the manufacturing domain and exploited the object-oriented principle of inheritance of attributes and methods in hierarchies.

The main components of SIMBIOPHARMA are depicted in Fig. 2. The system definitions comprise all the declarative and procedural knowledge for modelling manufacturing operations. The declarative knowledge consists of all the objects, such as tasks and resources, and their class definitions describing their properties. The procedural knowledge enables programmatic control over an application and takes the form of procedures, methods and rules. Each of these items contains G2 statements that can be used to obtain information about objects and to specify actions to be executed on them.

Some of the key features of the user interface are highlighted in the simplified schematic in Fig. 2. The graphical user interface

depicted simplifies rapid prototyping of specific cases since it is highly interactive and provides visualisation of the various manufacturing levels. The user interface provides a template of the manufacturing levels to guide the user through building their application graphically. Palettes provide the different types of tasks and resources required to populate a simulation model such that the building blocks can be cloned from the palette and placed on the appropriate level in the simulation model and configured. The resource pools in the plant are specified to describe the capabilities of the plant.

The task-oriented approach adopted requires that the task sequences within each recipe be generated and the resource requirements identified. Resource requirements were expressed explicitly for each task by using objects (*resource managers*) attached to the task to represent the association of a resource from a pool to the task. Attributes of the tasks and resources were initialised through the use of input/output tables. Whenever a simulation is run, the simulation clock advances with the completion of each task and the simulation model is animated to enable the user to view what is happening at any point in time. Dynamic simulation enables temporal reasoning and makes it possible to view the tasks that are active and to track the impact on parameters over time. Examples of such time-dependent parameters include the composition of process streams, the status of equipment, the product mass, the number of batches completed, resource availability and utilisation.

### 9.1. System definitions

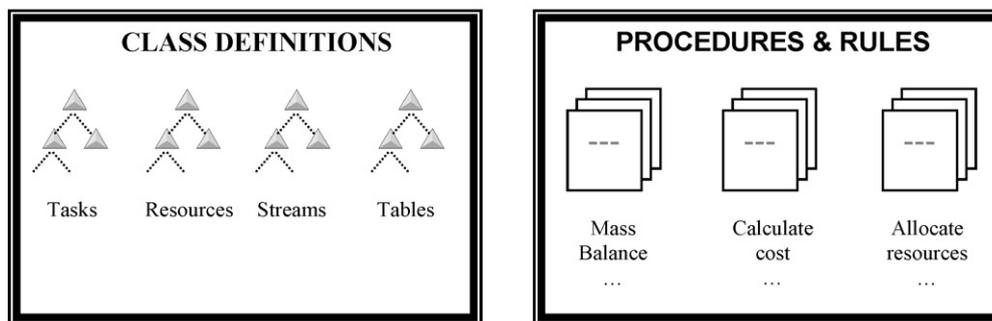
The three main constructs identified to specify a manufacturing process were the tasks carried out to manufacture a biopharmaceutical, the resources available within the plant, and the process streams that flow between material-handling tasks. Each type of construct required descriptions at multiple levels of detail represented as objects. The objects were categorised into classes defined by a set of attributes, which describe their properties, and methods, which characterise their functions. The constructs were classified into class hierarchies to allow inheritance of common properties and behaviour. Encapsulating the common features and behaviours of these objects into generic classes facilitated re-use of code and the extension of knowledge representation.

### 9.2. Manufacturing task classes

The tasks describing the manufacturing operations form the core of SIMBIOPHARMA. The hierarchical views of the manufacturing activities were realised through the use of workspaces, upon which items can be placed. Each high-level task on a workspace can be broken down into its subtasks that are placed on the subworkspace of the task. The manufacturing levels are therefore represented as a workspace hierarchy. This is illustrated in Fig. 3 where the tasks representing *product manufacture* each have an associated subworkspace with the constituent tasks that make up the recipe procedure.

The task class hierarchy is shown in Fig. 4 where the definitions of the tasks have been extended from ReThink's

SYSTEM DEFINITIONS



USER INTERFACE

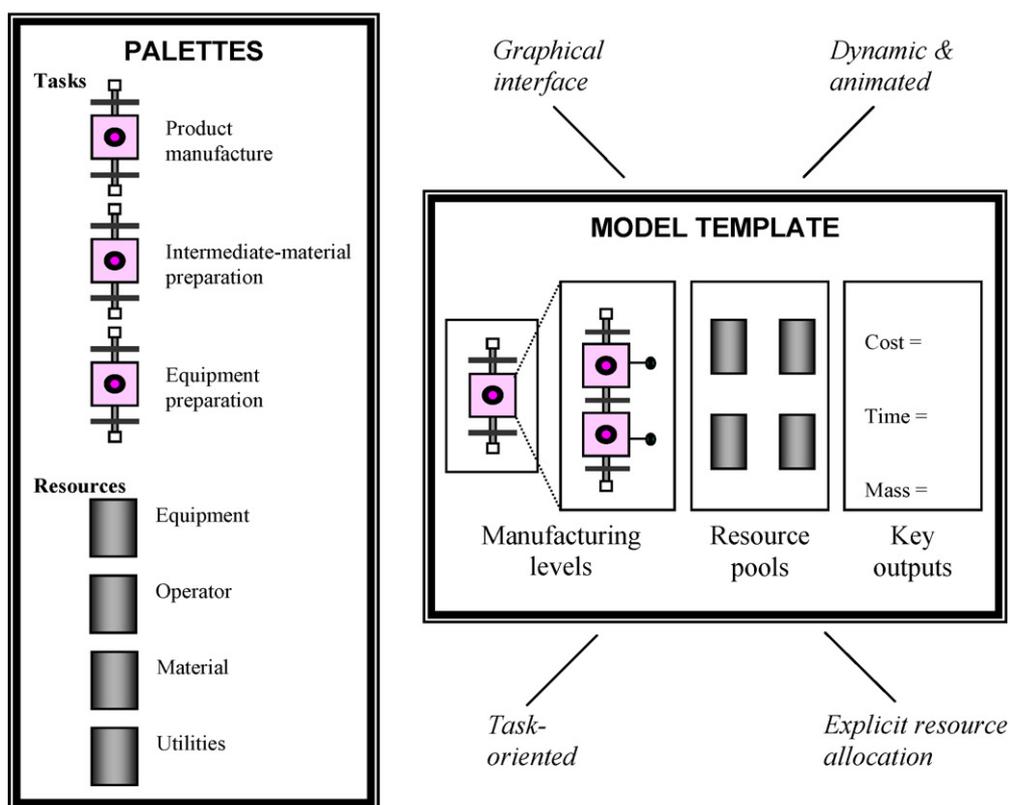


Fig. 2. Simplified schematic of the main components of the tool, SIMBIOPHARMA. The system definitions are comprised of class definitions, procedures and rules. Some of the features of the user interface are palettes and the model template.

system-defined class of a task block. The task blocks represent discrete events and have cost and duration attributes as well as methods defining default behaviour. These are all inherited by the custom tasks. When a task becomes active it calls methods to perform appropriate actions, such as cleaning or calculating stream compositions and activity costs.

An example of a *fermentation* task object and its definition is shown in Fig. 5. The class definition acts as a common template for all *fermentation* activities. It describes the common characteristics (attributes), behaviours (methods), and icon representation of every member of the *fermentation* class. For

the *fermentation* task, attributes are all inherited from superior classes, but a specific method is created to determine the composition of the outlet stream. *Fermentation* tasks are represented in simulation models by creating instances of the class, that appear graphically as icons, where each attribute is given a specific value. The attribute table indicates that information about the outlet streams, mass balance, duration and cost of the task need to be specified. When setting the duration of the task, the user can specify the duration to follow a particular distribution, such as a random normal distribution. The mean duration of the task and its standard deviation must then be provided. This feature

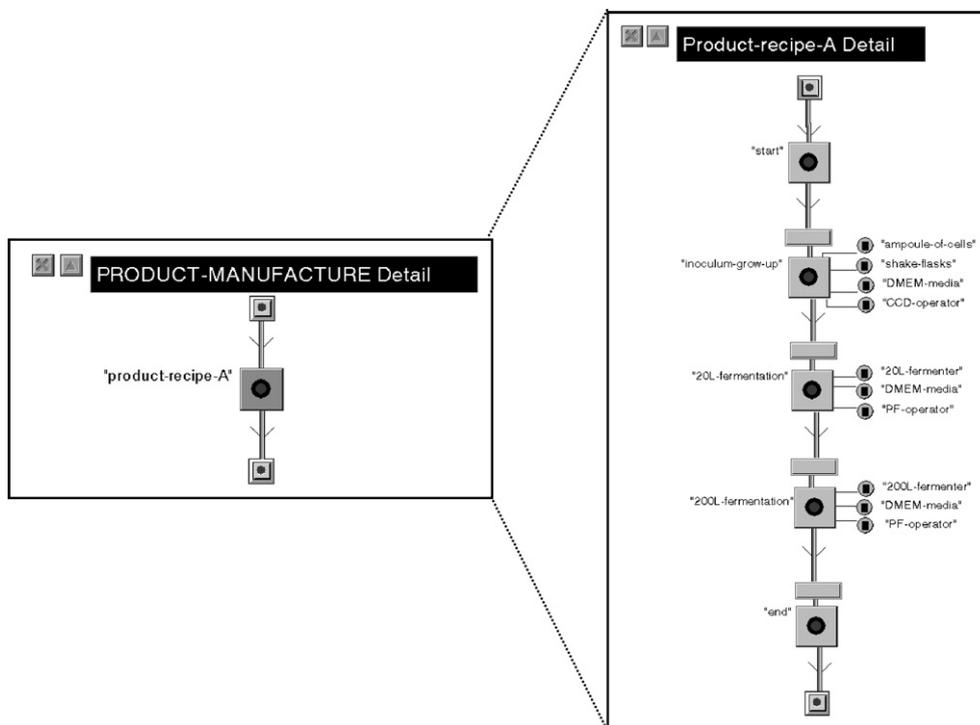


Fig. 3. Example of a workspace hierarchy. This shows tasks representing a product manufacture recipe and an equipment preparation recipe each with their associated subworkspaces.

allows variability in process durations to be captured. When the task is activated it calls the mass balance method as well as other methods defined for the superior classes to execute the required actions.

Regarding the mass balance inputs, the user may select different mass balance calculation methods for the *fermentation* task according to the level of sophistication desired. For example, at a certain level of detail it may be sufficient to determine the performance of unit operations using process models based on simple yield relations; whereas when more rigorous and predic-

tive evaluations are required, kinetic models may be selected. Each mass balance calculation method has an associated data table. The data table for a simple mass balance method for *fermentation* is illustrated in Fig. 5. This highlights that the simple mass balance method is based on the mass stoichiometry and the product titre being specified.

It was necessary to prevent tasks inheriting inappropriate attributes and methods. This was achieved by making a clear distinction between the overall hierarchical framework and the task class hierarchy (Fig. 4). So for example, the *campaign*

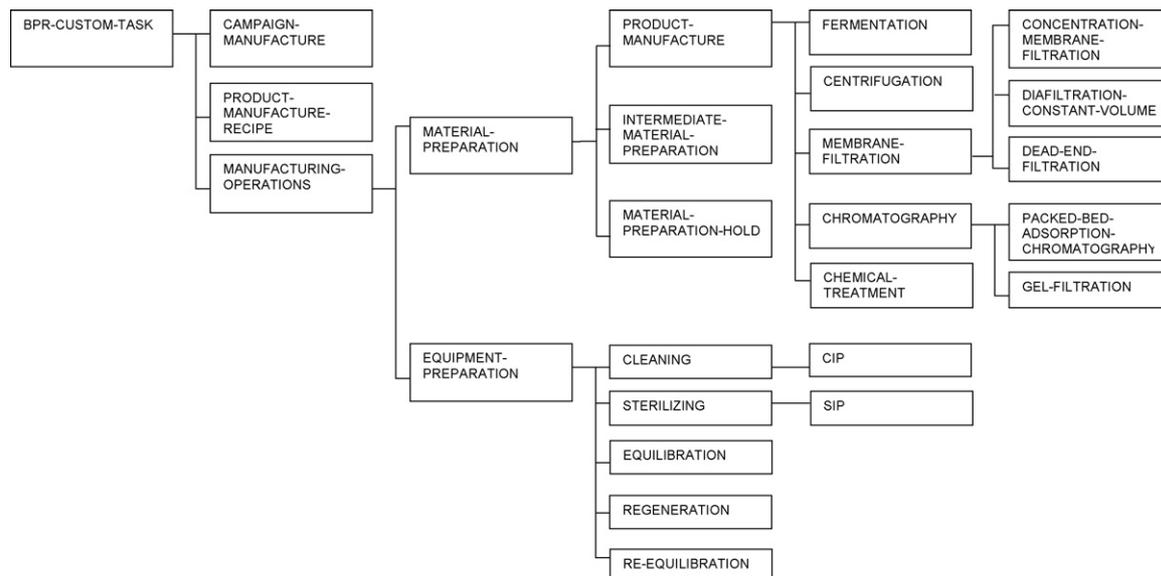


Fig. 4. Class hierarchy of key manufacturing tasks. The manufacturing operations are classified as either material preparation tasks or equipment preparation tasks.

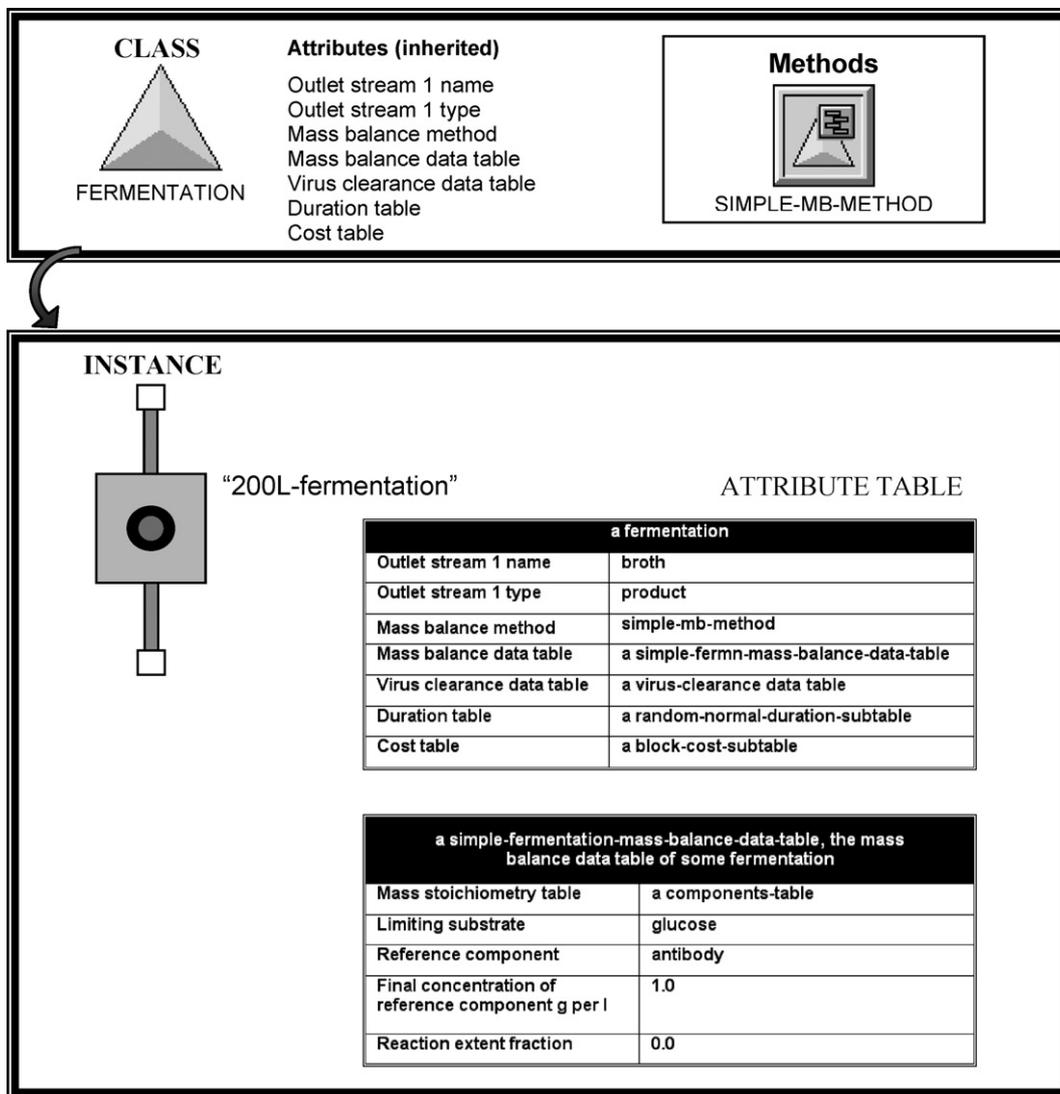


Fig. 5. Class definition and instance of fermentation task with its attribute table. The class definition acts as a common template for all fermentation tasks and the instance represents a specific occurrence of the task.

*manufacture* and *product manufacture recipe* tasks are classified separately from the remaining manufacturing operations. In addition, the biopharmaceutical manufacturing tasks are classified into *material preparation* and *equipment preparation* steps. The *campaign manufacture* task represents the top-level activity and its attributes include a “campaign summary table” indicating the total number of campaigns completed, cumulative product mass supplied and cumulative product mass demand. The *product manufacture recipe* task contains summary statistics on the progress of the batches for each campaign. Its attributes therefore include a “product-supply-and-demand-table” which monitor the number of batches completed, the product supplied per batch and the product demand per campaign.

### 9.2.1. Material preparation tasks

The *product manufacture* and *intermediate material preparation* tasks all share some characteristics and behaviours as they all make material. For this reason, the superior class, *material preparation*, was created to encapsulate the common informa-

tion into a single class. This minimises the use of redundant code, making the class hierarchy easier to comprehend, extend and maintain. The subclasses have more specific attributes to suit their function. For example, the *product manufacture* task block and its subclasses have attributes needed to perform mass balance calculations. Decoupling the mass balance data and associated calculation method from the task’s attributes was achieved by assigning the following attributes to the *product manufacture* definition: “mass-balance-method-name” and “mass-balance-data-table”. This meant that the subclasses, such as *fermentation* and *centrifugation*, could all inherit these attributes without any conflict. It also provided the flexibility to select different mass balance calculation methods for these tasks, and fill in their associated data requirements, according to the level of sophistication desired. To monitor the steps that achieve virus clearance, the *product manufacture* task block was also given the attribute “virus-clearance-data-table”, where it is possible to specify the “log-clearance-factor”. *Material preparation* tasks were also given attributes to specify the name and type

of outlet streams produced as a result of the activity. Examples of stream type are “product”, “intermediate” and “waste”.

### 9.2.2. Equipment preparation tasks

Equipment preparation tasks did not require specific attributes in addition to those inherited that include cost and duration tables. Modelling equipment preparation operations, such as *cleaning-in-place (CIP)* operations, was achieved at a simple level by creating a generic cleaning method for the equipment resource that changes the status of the equipment from “dirty” to “clean” and a *CIP* task block method that initially adjusts the equipment status to “cleaning-in-progress” and then invokes the equipment *CIP* method to perform the cleaning operation on the attached equipment. Similar methods were created for each of the equipment preparation operations.

### 9.2.3. Transitions

In addition to the manufacturing operations, it was necessary to create *transition* steps to be placed between the product manufacture tasks. These steps between tasks were created to serve two main purposes. Firstly, for discrete event simulation, the task duration and resource utilisations must be known before the task is activated. However, since certain operations, such as *filtration* and *chromatography* have methods to determine the task duration and volumes of buffer required, it was necessary to perform these calculations prior to starting these operations. The *transition* steps therefore call methods to compute these values and to set these attributes in the following task and its attached resources. The other fundamental function of the *transition* steps is to schedule the ancillary operations required between tasks. To this end the *transition* step has attribute tables, to specify the ancillary operations required prior to the next operation and post the previous operation, and methods to run the ancillary operations identified. For example a *chromatography* task will require the column to be cleaned and equilibrated before loading of the

process material can begin. Following elution, the column may require regeneration and another cleaning cycle, depending on its mode of use. Thus, the *transition* steps were key to achieving the necessary smooth functionality of the tool. They provided a mechanism for communication between the recipes and permitted the product manufacture recipe to drive the timing of the ancillary recipes.

### 9.3. Manufacturing resource classes

Modelling the resources that individual tasks require was achieved by defining resource pools, which represent a set of available resources, and resource managers, which associate a particular resource with a particular task in the simulation model.

The resources include the staff, equipment, materials and utilities required to execute the tasks and may also be generated as a result of a task execution, as in the case of *intermediate material preparation* tasks. The resource class hierarchy, depicted in Fig. 6, categorises the resources. Since ReThink already has definitions of resources, they were customised to reflect the nature of the current domain.

The resources were classified as renewable or non-renewable owing to their different behaviours. Renewable resources include *equipment* and *operators* since they are replaced in the resource pool after use. When a block begins processing, a resource manager allocates the required resource from the pool to the block, making it unavailable for other tasks. When the block finishes processing the resource is deallocated, making it available again for other tasks. In contrast, *materials* and *utilities* are consumed and generated by tasks, and are therefore classified as non-renewable resources.

#### 9.3.1. Operators and equipment

*Operators* inherited the properties and behaviour of renewable resources and did not require any further customisations

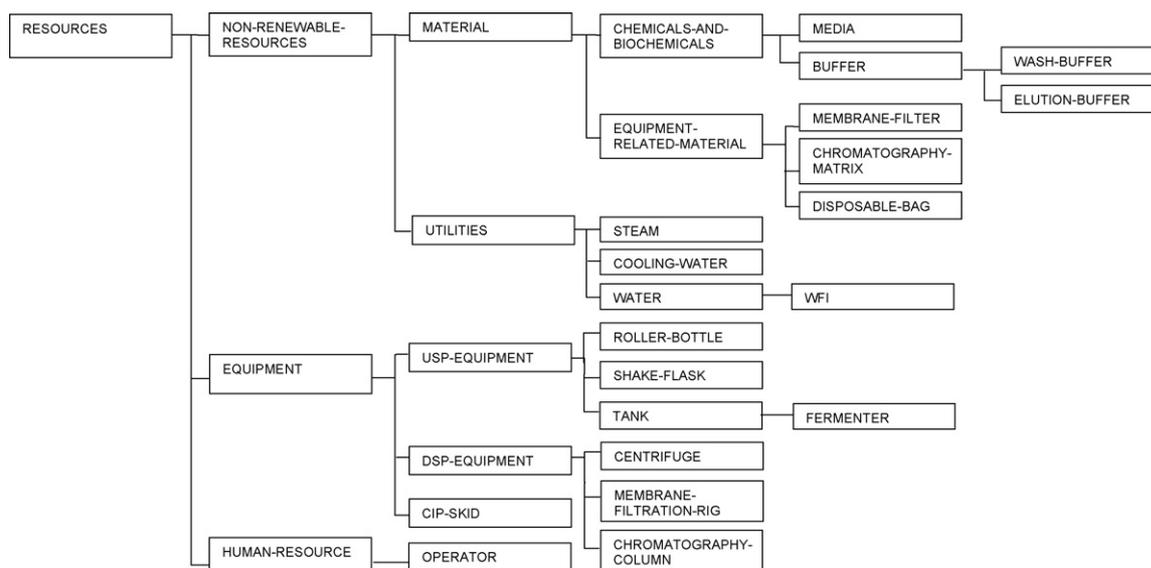
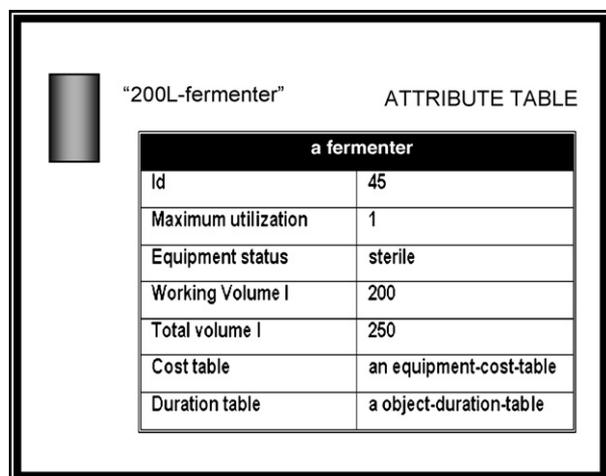


Fig. 6. Class hierarchy of manufacturing resources. These include non-renewable resources, e.g. material and utilities, and renewable resources, e.g. equipment and operators.



a fermenter	
Id	45
Maximum utilization	1
Equipment status	sterile
Working Volume l	200
Total volume l	250
Cost table	an equipment-cost-table
Duration table	a object-duration-table

Fig. 7. Example of the representation of a specific instance of a fermenter resource with its attribute table. This highlights the key properties that need to be specified for a fermenter.

to describe their cost and utilisation patterns. On the other hand, special features needed to be incorporated into the tool for *equipment* resources. They were given additional attributes, such as “equipment-status” (“clean”, “dirty” or “sterile”). More specific attributes indicating the equipment sizes were given to subclasses, such as *tank* and *chromatography column*. Fig. 7 shows an example of the representation of a specific instance of a *fermenter* resource with its attribute table.

The common operations for equipment were cleaning and sterilising activities. As mentioned earlier, generic “clean” and “sterilise” methods were created for the *equipment* class methods to change the status of equipment. Since objects are arranged in a class hierarchy, subclasses of equipment, such as fermenters, inherit this cleaning method. A key to the creation of this prototype tool was the use of inheritance of attributes and methods in class hierarchies as demonstrated above. This helped to minimise development efforts significantly. In addition chromatography columns were assigned methods for column preparation activities, such as “equilibrate”, “regenerate” and “re-equilibrate”. These alter the status of columns when invoked; to “equilibrated” for example.

The status attribute provides an additional constraint on the availability of equipment and permits tasks to specify the equipment status they require to function. For example, a *fermentation* task block requiring a “sterile” *fermenter* will only become active if the *fermenter* resource allocated to it has the correct status. Allocation of equipment resources was controlled by specification of the “required-initial-equipment-status” for each *equipment-resource-manager* attached to a task and a procedure to choose a resource from the pool with the correct status.

### 9.3.2. Materials and utilities

Unlike equipment and personnel resources where the utilisation refers to their number, for non-renewable resources the utilisation can refer to a specific attribute in each case. An efficient approach to solving this problem was to assign material

resources the attribute “utilisation-indicator”. This permits the use of one generic rule and one generic method for all material resources, which work together to update the attribute named by the “utilisation-indicator”. Creating generic rules and methods enhances the reusability of the application and avoids duplication of code. This approach avoids ambiguity that can result from not knowing exactly which attribute is used to indicate the resource’s utilisation and provides much more flexibility for configuring individual material and utilities resources within a large application.

Material resources were classified into *chemicals-and-biochemicals*, e.g. glucose, and *equipment-related-material*, e.g. membrane-filter. The former was assigned a physical state attribute, composition tables for component masses, volumes, concentrations and mass fractions and a virus table detailing the virus titre in the resource. Each subclass of equipment-related-material was assigned attributes to indicate its size, e.g. “area” and “number-of-units” for membrane-filters.

In biopharmaceutical facilities, consumables, such as membranes and matrices, can be re-used for a certain lifetime or be disposed of after each batch/use. To distinguish between these two types of uses of equipment-related material, the *equipment-related material* class was given the attribute “re-use”, that could be true or false, to indicate whether the material could be re-used in subsequent batches. To ensure that costs of disposable materials were not duplicated if the materials were used in multiple cycles of a step or in ancillary activities as well as the main activity, the *equipment resource manager* was given the attribute “dispose-after-task” that could be true or false to indicate whether the material attached to a particular task was immediately disposed of upon task completion or used by another task before disposal. The ability to model flexibly resources along with their corresponding usage patterns is a key requirement in most manufacturing applications.

### 9.4. Process streams

The process streams refer to the flows of material out of product manufacture operations and intermediate material preparation operations. Examples include the culture broth produced by a fermentation task, and fermentation media produced by a media preparation task. The *process stream* object was given attributes to specify its type and define its composition in terms of the component masses, volumes, concentrations, mass fractions, and their respective totals. In addition, the number of virus particles in the stream was monitored.

### 9.5. Cost objects

The framework, illustrated in Fig. 1, led to the identification and explicit modelling of new cost items incurred from ancillary activities, e.g. cleaning equipment. In traditional costing techniques, these costs may be hidden in overheads or unaccounted for. The cost of each manufacturing task was based on the costs incurred through the use of the allocated resources (excluding equipment). Hence, the framework provided a clear linkage between manufacturing activities and their associated

direct resource costs. Cost tables were created for the fixed capital investment and the cost of goods. The attributes of each table were configured to hold input and output values. For the fixed capital investment, the input attribute assigned was the “Lang factor”, and the output ones were the “total equipment purchase cost” and the “fixed capital investment”. For the cost of goods table, it was necessary to specify the depreciation period and the facility size. The outputs in the table included cost categories, such as raw materials, utilities, staff, overheads and depreciation. The cost of each task is updated based on the resources consumed. Each table was assigned methods to compute the individual cost items, the total fixed capital investment and cost of goods per gram. The cost methods, associated with cost of goods table, compute the direct costs based on the utilisation of the material, operator and utilities resources. The fixed facilities overhead costs are derived from the capital investment and the processing time.

### 9.6. Representing uncertainties in parameter values

ReThink only supports the representation of probability distributions for the duration of blocks. However, other parameter values are given single values with no option of specifying a distribution to express the uncertainty in the parameter value. The method proposed to handle stochastic parameters was to use discrete distributions where each possible value of a parameter was given a probability of occurrence. This was represented using ReThink’s *branch* blocks with several output paths leading to blocks representing the different outcomes.

### 9.7. Reporting

Task blocks and associated procedures were created to write the outputs of each campaign and Monte Carlo simulation to a file that could be viewed in Microsoft Excel. This facilitated the generation and storage of report files. The strengths of Excel for producing graphs and performing statistical analysis of the data could then be exploited. This was particularly useful for handling the extensive amount of data generated from running Monte Carlo simulations. Suitable graphical charts were proposed to provide greater insight into the process-business interactions and to ensure the results of such analyses were readily interpretable. These were used to help determine the ranking of alternatives under different scenarios. These included tornado diagrams, risk versus reward charts, bubble plots and windows of operation. In addition resource utilisation plots could be generated in ReThink to assess whether modifications to operations would help reduce the frequency of peaks in demand and provide a more streamlined process.

## 10. Using the prototype tool

The object-oriented representation makes it easier to transform knowledge about a process rapidly into a graphical model that is easy to understand and use. The user generates simulation models of a manufacturing process by simply cloning the customised objects (e.g. *fermenter* resources, *fermentation*

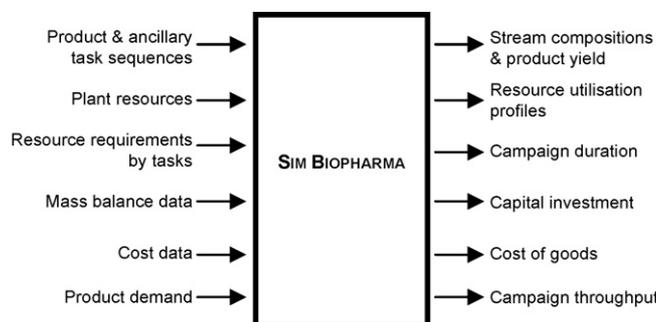


Fig. 8. Inputs to and outputs from the tool, SIMBIOPHARMA. The key inputs are the tasks, resources, costs and process data. The key performance measures are yields, resource utilisation, and investment costs, operating costs and processing times per campaign.

tasks) from palettes and dropping them onto workspaces. The objects’ attributes are then configured for the specific case and the task blocks connected together to create a running simulation model. A manufacturing template was also built to guide the user through building a simulation model of the manufacturing system under investigation. Fig. 2 illustrates part of the template showing the workspace hierarchy as well as the multiple resource pools and a summary of the key outputs. The key inputs to and outputs from SIMBIOPHARMA are summarised in Fig. 8.

After a particular case is set up the impact of different production strategies on the process performance, resource utilisation and bottlenecks, and the resultant cost of goods and risks can be evaluated. Examples of some of the results from a case study set up to assess the process economics of a pilot plant based on disposable components as opposed to stainless steel for clinical trial material preparation are shown in Figs. 9–11. Fig. 9 shows the combined media and buffer utilisation over time for each pilot plant for both the product manufacture recipe and the equipment preparation recipes (CIP, SIP). Here media refers to fermentation media and buffer refers to the solutions used during the equilibration, wash and elution stages of chromatography and to the CIP solutions. The figures highlight the greater demand on buffer in a stainless steel pilot plant that can be attributed to the need for CIP procedures in addition to the tasks involved in the product manufacture recipe. A simplification for the case study was that media and buffer arrived ready-made to each plant. This is certainly appropriate for the disposables-based plant since they could arrive pre-sterilised in bags ready for use. However, in a stainless steel plant, it is more likely that these materials would be prepared in-house. Consequently, the utilisation curves indicate that more media and buffer preparation steps are required in a stainless steel plant.

Operator utilisation charts were also generated and they would enable a company to determine the appropriate resource pool (operators) suitable for carrying out the manufacturing tasks efficiently. Alternatively the current utilisation data allows the company to identify periods during manufacturing when an operator resource is under-utilised and can be freed up for other tasks. The utilisation metrics therefore provide insight into how efficiently the process is using the operators and can help prompt

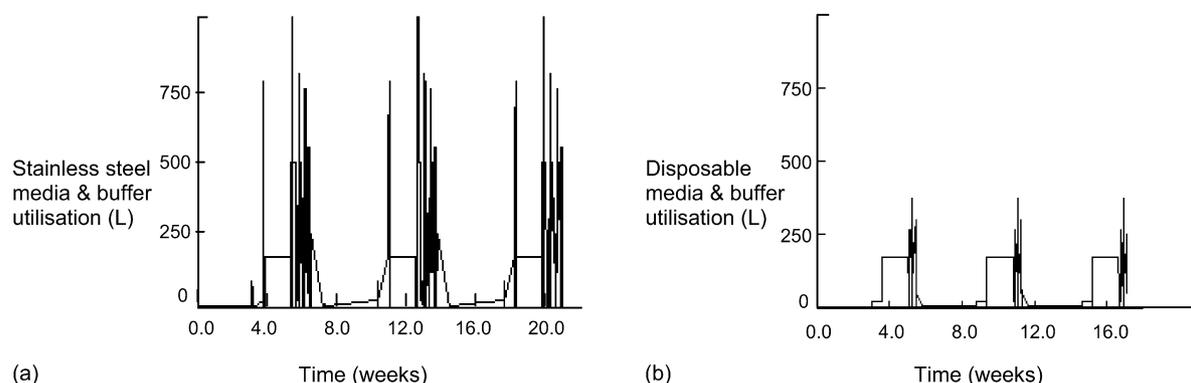


Fig. 9. Combined utilisation of media and buffer over time for (a) the stainless steel pilot plant and (b) the disposables-based pilot plant. The peaks in the stainless steel plant correspond to the high buffer demands in cleaning-in-place (CIP) steps, which are absent in the disposables-based plant.

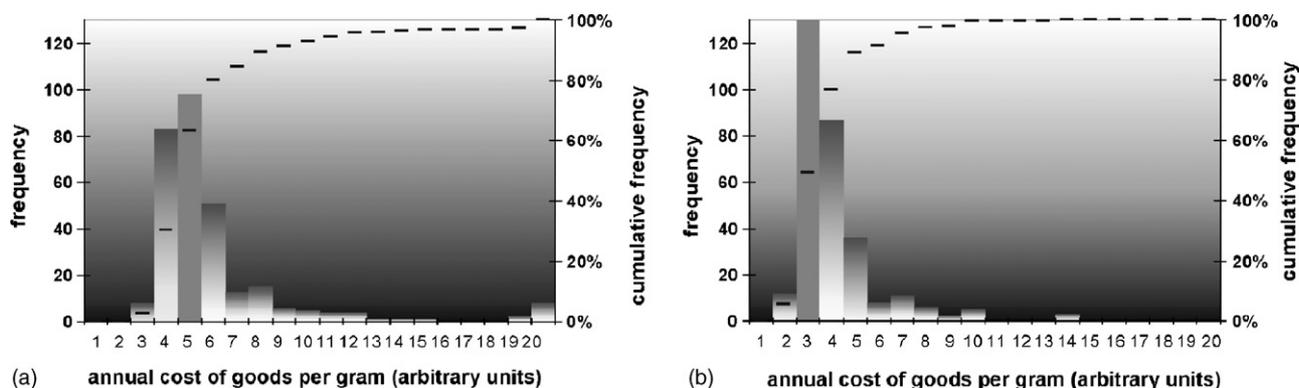


Fig. 10. The frequency distribution and cumulative frequency for the annual cost of goods per gram for (a) the stainless steel plant and (b) the disposables plant. The solid column indicates the deterministic estimate of the cost.

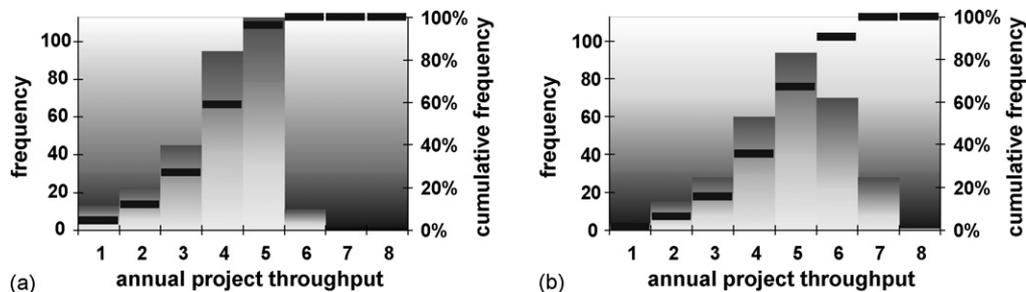


Fig. 11. The frequency distribution and cumulative frequency for the annual project throughput for (a) the stainless steel plant and (b) the disposables plant.

reorganisation of resources and tasks to maximise resource utilisation so as to improve throughput and productivity.

In this example case study, Monte Carlo simulations were used to characterise the variability in the key performance measures due to uncertainties in product titre, product demand and market success and examine the impact on this facility decision. Examples of the financial and throughput outputs from this case study are shown in Figs. 10 and 11. The range in possible outcomes for the performance measures highlights the danger of using deterministic outputs for decision-making. From the COG/g outcomes in Fig. 10, it is possible to conclude that the disposables plant has both a lower expected cost and lower risk as the outcomes are less disperse than for the stainless steel plant.

The disposables plant also offers a greater likelihood of achieving an annual throughput of at least four projects (84% versus 73%) as can be seen in Fig. 11.

## 11. Conclusions

The research, programming and decisions made in order to construct a prototype software tool, SIMBIOPHARMA, has been presented. SIMBIOPHARMA, provides a hierarchical framework for modelling both the technical and business perspectives of biopharmaceutical batch manufacturing processes. SIMBIOPHARMA can describe a large number of problems facing the industry, such as facility, process and capacity decisions. The

hierarchical approach to representing activities confers maximal flexibility since it allows production strategies to be evaluated at different levels of detail according to the goals of the user. The object-oriented representation of the key objects, their properties and behaviour provides full access to the base language and hence facilitates incorporation of new or changed requirements once implementation is underway. It also encourages the development of well-modularised applications. The tool combines interactive graphics, animation, and dynamic simulation to create a more flexible environment for modelling processes than that found in conventional process software tools. It adopts a task-oriented approach to modelling manufacturing and incorporates the interactions between manufacturing tasks (e.g. fermentation, chromatography) and the available resources (e.g. manpower, materials, equipment). The task-oriented approach also promotes transparency, as it is possible to view the cost, duration and yield of each task. The tool also encourages the explicit modelling of ancillary tasks, e.g. equipment cleaning-in-place (CIP) and hence the cost models account for items that might otherwise be overlooked. In addition, risk and uncertainty are dealt with and so investment appraisal can be based on both the expected outputs and the likelihood of achieving certain threshold values. The performance measures built into the tool to evaluate manufacturing alternatives can be

summarised in terms of cost, time, mass, resource utilisation and risk. This framework has been used for a series of case studies ranging from facility decisions (Farid, Novais, Karri, Washbrook, & Titchener-Hooker, 2000; Farid, Washbrook, & Titchener-Hooker, 2005a,b), to process decisions (Mustafa et al., 2004) and capacity decisions (Lim et al., 2005; Lim, Zhou, Washbrook, Titchener-Hooker, & Farid, 2005). They serve to illustrate how the prototype application can be used to assess manufacturing strategies in terms of their operational benefits and cost-effectiveness.

### Acknowledgements

Financial support from the Biotechnology and Biological Sciences Research Council (BBSRC) and Lonza Biologics is gratefully acknowledged. The Advanced Centre for Biochemical Engineering at UCL houses the Innovative Manufacturing Research Centre (IMRC) and collaborates with a range of academic partners and biopharmaceutical and biotechnology companies.

### Appendix A. Key mass balance models in SIMBIOPHARMA

Inputs	Outputs	Mass balance calculations
<b>Fermentation</b>		
Mass-stoichiometry coefficient, $a_i$	Outlet stream component mass, $m_{i \text{ out}}$	1. $m_{\text{out ref}} = c_{\text{ref}} V_{\text{in tot}}$
Feed stream component mass, $m_{i \text{ in}}$	Extent of reaction, $x$	2. $x = \frac{m_{\text{in ref}} - m_{\text{out ref}}}{m_{\text{in jim}}} \frac{a_{\text{im}}}{a_{\text{ref}}}$
Reference component, ref		3. $m_{i \text{ out}} = m_{i \text{ in}} - m_{\text{in jim}} x \frac{a_i}{a_{\text{im}}}$
Final concentration of ref component, $c_{\text{ref}}$		
Total volume of feed stream, $V_{\text{in tot}}$		
Limiting substrate, lim		
<b>Centrifugation</b>		
Solids-carry-over-fraction, $S$	Solids removal fraction, $R$	1. $R = 1 - S$
Solid-volume-fraction in sediment, $v_{\text{s sed}}$	Supernatant and sediment solid component masses, $m_{\text{s sup}}, m_{\text{s sed}}$	2. $m_{\text{s sed}} = R m_{\text{s in}}$ $m_{\text{s sup}} = m_{\text{s in}} - m_{\text{s sed}}$
Total feed stream component masses, $m_{\text{in tot}}$	Supernatant and sediment liquid component masses, $m_{\text{li sup}}, m_{\text{li sed}}$	3. $m_{\text{li in}} = m_{\text{in tot}} - m_{\text{s in}}$ $V_{\text{in}} = V_{\text{in tot}} - V_{\text{s in}}$ $\rho_{\text{li in}} = \frac{m_{\text{li in}}}{V_{\text{in}}}$
Total volume of feed stream, $V_{\text{in tot}}$		4. $V_{\text{s sed}} = \frac{m_{\text{s sed}}}{\rho_{\text{s}}}$ $V_{\text{tot sed}} = \frac{V_{\text{s sed}}}{v_{\text{s sed}}}$ $V_{\text{li sed}} = V_{\text{tot sed}} - V_{\text{s sed}}$ $m_{\text{li sed}} = V_{\text{li sed}} \rho_{\text{li in}}$
Solid density, $\rho_{\text{s}}$		5. $m_{\text{li sed}}, m_{\text{li sup}}$ $m_{\text{li sed}} = \frac{m_{\text{li in}}}{m_{\text{li in}}} m_{\text{li sed}}$ $m_{\text{li sup}} = m_{\text{li in}} - m_{\text{li sed}}$
*Assumption: only 1 solid component		
<b>Membrane filtration</b>		
Calculation mode:	Output mode:	1. $A = \frac{V_{\text{in}}(1 - \text{CF}^{-1})}{JnA}$ or $t = \frac{V_{\text{in}}(1 - \text{CF}^{-1})}{JnA}$
Membrane area per unit, $A$ /Processing time per cycle, $t$	Membrane area per unit, $A$ /Processing time per cycle, $t$	2. $T = nt$
Average flux, $J$	Total processing time, $T$	3. $V_{\text{ret}} = \frac{V_{\text{in}}}{\text{CF}}$

Inputs	Outputs	Mass balance calculations
Volume of feed stream, $V_{in}$	Permeate and retentate stream component masses, $m_{i\text{ perm}}$ , $m_{i\text{ ret}}$	4. $m_{i\text{ retRC}\neq 0} = m_{i\text{ in}} \text{CF}^{(\text{RC}_i - 1)}$ $V_{i\text{ retRC}\neq 0} = \frac{m_{i\text{ retRC}\neq 0}}{\rho_i}$
No. of cycles, $n$		5. $V_{\text{retRC}=0} = V_{\text{ret}} - \sum V_{i\text{ retRC}\neq 0}$ $V_{i\text{ retRC}=0} = \frac{V_{i\text{ in}}}{V_{\text{inRC}=0}} V_{\text{retRC}=0}$ $m_{i\text{ retRC}=0} = V_{i\text{ retRC}=0} \rho_i$
Concentration factor, CF Rejection coefficient, RC		6. $m_{i\text{ perm}} = m_{i\text{ in}} - m_{i\text{ ret}}$
<b>Chromatography</b>		
Column height, $H$	Processing time per cycle, $t$	1. $t_r, r = L, W, E, t_r = \frac{HCV_r}{u_r}$
No. of column volumes, $CV$	Buffer volumes required, $V$	2. $V_r = CV_r V_{\text{col}} n$
Linear flow rates, $u$	Product and waste stream masses, $m_{i\text{ prod}}$ , $m_{i\text{ waste}}$	3. $V_{\text{prod}} = CV_{\text{prod}} V_{\text{col}}$ $m_{i\text{ prod from L}} = m_{i\text{ L}} y_i$ $V_{i\text{ prod from L}} = \frac{m_{i\text{ prod from L}}}{\rho_i}$ $V_{\text{prod from E}} = V_{\text{prod}} - \sum V_{i\text{ prod from L}}$ $V_{i\text{ prod from E}} = \frac{V_{i\text{ E}}}{V_{\text{E}}} V_{\text{prod from E}}$ $m_{i\text{ prod from E}} = V_{i\text{ prod from E}} \rho_i$ $m_{i\text{ prod}} = m_{i\text{ prod from L}} + m_{i\text{ prod from E}}$ $m_{i\text{ waste}} = m_{i\text{ L}} + m_{i\text{ W}} + m_{i\text{ E}} - m_{i\text{ prod}}$
Column volume, $V_{\text{col}}$		
No. of cycles, $n$		
Yield fraction, $y$		
Product-stream-column-volumes, $CV_{\text{prod}}$		
<b>Viral clearance</b>		
No of virus units in inlet stream, $vr_{in}$	No of virus units in outlet stream, $vr_{out}$	1. $vr_{out} = vr_{in} 10^{\text{LCF}}$
Log clearance factor, LCF		

## References

- Atkinson, B., & Mavituna, F. (1991). *Biochemical engineering and biotechnology handbook* (2nd ed.). Basingstoke: Macmillan. New York, N.Y: Stockton Press. Chapter 19.
- Banks, J. (1998). Principles of simulation. In J. Banks (Ed.), *Handbook of simulation: Principles, methodology, advances, applications, and practice* (pp. 3–30). Chichester: Wiley, Engineering & Management Press.
- Bastiaan, H. K. (1998). Process model and recipe structure. The conceptual design for a flexible batch plant. *ISA Transactions*, 36(4), 249–255.
- Bernstein, D. F., & Hamrell, M. R. (2000). Integrating drug supply issues with strategic preclinical and clinical development. *Drug Inf. J.*, 34, 909–917.
- Bogle, I. D. L., Cockshott, A. R., Bulmer, M., Thornhill, N., Gregory, M., & Dehgant, M. (1996). A process systems engineering view of biochemical process operations. *Comput. Chem. Eng.*, 20(6–7), 943–949.
- Byrom, D. (2000). Role and timing of process development for biopharmaceutical manufacture. *Pharm. Technol. Eur. (BioPharm Eur.)*, 12(3), 52–56.
- Chadd, H. E., & Chamow, S. C. (2001). Therapeutic antibody expression technology. *Curr. Opin. Biotechnol.*, 12, 188–194.
- Clemento, A. (1999). New and integrated approaches to successful accelerated drug development. *Drug Inf. J.*, 33, 699–710.
- Copmann, T., Davis, G., Garnick, R., Landis, J., Lubiniecki, A., Massa, T., et al. (2001). One product, one process, one set of specifications: A proven quality paradigm for the safety and efficacy of biologic drugs. *BioPharm.*, 14–24.
- Crowl, T. E. (1997). S88.01 Concepts streamline control software application for biotech plant. In *ISA Tech/Expo Technology Update Conference Proceedings*, 1 (2) (pp. 131–141).
- Datar, R. V., Cartwright, T., & Rosen, C. (1993). Process economics of animal cell and bacterial fermentations: A case study analysis of tPA. *BioTechnology*, 11, 349–357.
- Farid, S. S., Washbrook, J., & Titchener-Hooker, N. J. (2005a). Decision-support tool for assessing bio-manufacturing strategies under uncertainty: Stainless steel versus disposable equipment for clinical trial material preparation. *Biotechnol. Prog.*, 21(2), 486–497.
- Farid, S. S., Washbrook, J., & Titchener-Hooker, N. J. (2005b). Combining multiple quantitative and qualitative goals when assessing biomanufacturing strategies under uncertainty. *Biotechnol. Prog.*, 21(4), 1183–1191.
- Farid, S., Novais, J. L., Karri, S., Washbrook, J., & Titchener-Hooker, N. J. (2000). A tool for modelling strategic decisions in cell culture manufacturing. *Biotechnol. Prog.*, 16(5), 829–836.
- Gosling, I. (1996). The role of computer simulation in the bioprocessing industry. *BioPharm. Jan.*
- Gregersen, J. (1995). Biomedical product development. In H.-J. Rehm, G. Reed, A. Puhler, P. Stadler, & D. Brauer (Eds.), *Biotechnology: Legal, economic and ethical dimensions vol. 12*, (2nd ed., pp. 213–237). Weinheim: VCH.
- Gritsis, D., & Titchener-Hooker, N. J. (1989). Biochemical process simulation. In *ICHEME Symposium Series*, 114 (pp. 69–77).
- Hamers, M. N. (1993). Multiuse biopharmaceutical manufacturing. *BioTechnology*, 561–570. May.
- Hayes, R. H., & Wheelwright, S. C. (1984). *Restoring our competitive edge: Competing through manufacturing*. New York: Chichester: Wiley.
- Johnsson, C., & Arzen, K.-E. (1998). Grafchart for recipe-based batch control. *Comput. Chem. Eng.*, 22(12), 1811–1828.
- Karri, S., Davies, E., Titchener-Hooker, N. J., & Washbrook, J. (2001). Biopharmaceutical process development: Part III A framework to assist decision making. *Biopharm Eur.*, 76–82.
- Kelley, B. D. (2001). Bioprocessing of therapeutic proteins. *Curr. Opin. Biotechnol.*, 12(2), 173–174.
- Knutilla, A., Schlenoff, C., Ray, S., Polyak, S. T., Tate, A., Cheah, S. C., et al. (1998). *Process specification language: An analysis of existing representa-*

- tions. *NISTIR 6160*. Gaithersburg, MD: National Institute of Standards and Technology.
- Lakshmanan, R., & Stephanopoulos, G. (1998). Synthesis of operating procedures for complete chemical plants – I. Hierarchical structures modeling for non-linear planning. *Comput. Chem. Eng.*, *12*(9), 985–1002.
- Lang, H. J. (1948). Simplified approach to preliminary cost estimates. *Chem. Eng.*, *55*, 112–113.
- Lim, A. C., Zhou, Y., Washbrook, J., Sinclair, A., Fish, B., Francis, R., et al. (2005). Application of a decision-support tool to assess pooling strategies in perfusion culture processes under uncertainty. *Biotechnol. Prog.*, *21*(4), 1231–1242.
- Lim, A. C., Zhou, Y., Washbrook, J., Titchener-Hooker, N. J., & Farid, S. S. (2005). A computer-aided approach to compare the production economics of fed-batch versus perfusion culture. *Biotechnol. Bioeng.*, *93*(4), 687–697.
- Manivannan, M. S. (1998). Simulation of logistics and transportation systems. In J. Banks (Ed.), *Handbook of simulation: Principles, methodology, advances, applications, and practice* (pp. 571–604). Chichester: Wiley Engineering & Management Press.
- Mison, D., & Curling, J. (2000). The industrial production costs of recombinant therapeutic proteins expressed in transgenic corn. *BioPharm.*, 48–54. May.
- Moilanen, T., & Martin, C. (1996). *Financial evaluation of environmental investments*. Rugby, England: Institution of Chemical Engineers.
- Mustafa, M. A., Washbrook, J., Lim, A. C., Zhou, Y., Titchener-Hooker, N. J., & Morton, P., et al. (2004). A software tool to assist business-process decision making in the biopharmaceutical industry. *Biotechnol. Prog.*, *20*(4), 1096–1102 [Correction: *Biotechnol. Prog.*, (2005), *21*(1), 320].
- Nelson, K. (1998). Cell culture manufacturing: An economic perspective from an engineering consultant. In *US Biotechnology Symposium*.
- Novais, J. L., Titchener-Hooker, N. J., & Hoare, M. (2001). Economic comparison between conventional and disposables-based technology for the production of biopharmaceuticals. *Biotechnol. Bioeng.*, *75*(2), 143–153.
- Osborne, A. (1997). *Evaluation of project economic viability*. UCL, UK: Presented at MBI Training Programme: Design II. May.
- Peters, M. S., & Timmerhaus, K. D. (1991). *Plant design and economics for chemical engineers*. London, New York: McGraw-Hill. Chapter 6.
- Petrides, D. P., & Sapidou, E. S. (1997). Optimization of protein recovery using computer-aided process design tools. In E. Goldberg (Ed.), *Handbook of downstream processing*. London, New York: Blackie Academic & Professional.
- Petrides, D. P., Calandranis, J., & Cooney, C. (1995). Computer-aided design techniques for integrated biochemical processes. *Genet. Eng. News. Sep.*, *10*(28–29), 35.
- Pisano, G., & Wheelwright, S. C. (1995). High-tech R&D. *Harvard Business Review*, 93–105.
- Polakovic, M., & Mandenius, C. F. (1996). Simulation of a complex biotechnological process using an object-oriented knowledge-based system. In A. R. Moreira, & K. K. Wallace (Eds.), *Computer and information science applications in bioprocess engineering* (pp. 157–163). Dordrecht, London: Kluwer in cooperation with NATO Scientific Affairs Division.
- Pugh, G. (1998). Cell culture manufacturing: An economic perspective from a contract manufacturer's viewpoint. In *US Biotechnology Symposium*.
- Puigjaner, L., & Espuna, A. (1998). Prospects for integrated management and control of total sites in the batch manufacturing industry. *Comput. Chem. Eng.*, *22*(1–2), 87–107.
- Rajapakse, A., Titchener-Hooker, N. J., & Farid, S. S. (2005). Modelling of the biopharmaceutical drug development pathway and portfolio management. *Comput. Chem. Eng.*, *29*(6), 1357–1368.
- Sadana, A., & Beelaram, A. M. (1994). Efficiency and economics of bioseparation: Some case studies. *Bioseparation*, *4*, 221–235.
- Savage, C. (2000). Scaleup and bioproduction. *Genet. Eng. News*, *1*(65), 80.
- Shanklin, T., Roper, K., Yegneswaran, P. K., & Marten, M. R. (2001). Selection of bioprocess simulation software for industrial applications. *Biotechnol. Bioeng.*, *72*(4), 483–489.
- Sinnott, R. K. (1993). *Coulson and Richardson's chemical engineering (chemical engineering design)* (Vol. 6, pp. 209–244). Oxford: Pergamon Press.
- Sofer, G., & Hage, L. (1997). *Handbook of process chromatography: A guide to optimization, scale-up, and validation*. San Diego, CA, London: Academic Press. Chapter 10.
- Stanley, G. M. (1994). The emerging trend towards knowledge-based frameworks for computer-integrated manufacturing. In *Instrument Society of America ISA/94 Conference*.
- Stephanopoulos, G., Ali, S., Linninger, A. A., & Salomone, E. (2000). Batch process development: Challenging traditional approaches. In *AICHE Symposium Series*, *96*(323), 46–57.
- Subrahmanyam, S., Pekny, J. F., & Reklaitis, G. V. (1996). Decomposition approaches to batch plant design and planning. *Ind. Eng. Chem. Res.*, *35*, 1866–1876.
- Varga, E. G., Titchener-Hooker, N. J., & Dunnill, P. (2001). Prediction of the pilot-scale recovery of a recombinant yeast enzyme using integrated models. *Biotechnol. Bioeng.*, *74*(2), 96–107.
- Zhou, Y. H., Holwill, I. L. J., & Titchener-Hooker, N. J. (1997). A study of the use of computer simulations for the design of integrated downstream processes. *BioProcess Eng.*, *16*, 367–374.