Business process design: flexible modelling with multiple levels of detail

Lynne P. Baldwin, Tillal Eldabi and Ray J. Paul

Department of Information Systems and Computing,
Brunel University, Uxbridge, UK

Abstract

Purpose Most manufacturing processes tend to involve more than one level of detail at the design phase. These often consist of a higher level that represents the building blocks of the firm and a lower level that represents a more detailed structure of the process. When designing such processes, this type of structure is difficult to capture without some form of modelling. In such cases simulation can be used to help overcome this problem. This paper presents an investigation of simulation packages.

Design/methodology/approach These simulation packages were investigated regarding their abilities to model business processes related to manufacturing systems.

Findings The research findings suggest that no one simulation package currently available can alone offer sufficiently flexible facilities for the variable detailed modelling of manufacturing systems design.

Originality/value - The paper relates to one specific design framework called manufacturing system design (MSD). It defines the higher level of detail as the conceptual modelling level and the lower level as the detailed design level. A four-step framework is proposed, and it is argued that this may better deal with problems of detail variability.

Keywords Business process re-engineering, Simulation, Computer software

Paper type Research paper

1. Introduction

In most business processes, the design stage is vital, whether for new or re-engineered systems. Most business process design involves more than one level of detail. There are essentially two levels: a higher level which represents the building blocks of the particular process, and a lower level that represents everyday operations. The lower level also includes extra details, such as the physical location of the components that constitute the process. In business process re-engineering (BPR), the organisation actually abandons the old process and builds a completely new one (Hammer and Champy, 1994; Blacker, 1995). Such processes, with multiple levels of detail, are often complex. For designing this type of process, some form of modelling is required that will enable decision makers to select the most suitable configuration to produce the desired output. Business process modelling (BPM) in general is one important activity throughout the design process for both levels of detail. BPM has proved useful in BPR when a radical change is proposed, as analysts need to have a clear overview of the system prior to implementation (Ould, 1995). Simulation is one type of modelling, and it offers many benefits in BPR (Bhaskar et al., 1994; Giaglis et al., 1999; Lee and Elen, 1996).

This paper addresses the use of simulation software in BPM, with particular reference to the manufacturing process. It describes research on the ability of existing
simulation software to offer variable detail BPM during the design process. In order to choose the right simulation package for design purposes, it is first necessary to study the capability of simulation software in modelling the higher level of detail in the business process. The next issue is how well the package can model the subsequent details in such a way as to avoid repeating the modelling and data collection effort already gathered at the higher level. The manufacturing systems design process is divided into two levels of detail, the conceptual modelling level and the detailed design level. The first, higher level concerns the development of the basic principles by which the system will work. The second, lower level provides a detailed account of what is required (Wu, 1992a). Simulation is considered as an increasingly important computer aid to the design process, partly because of the growing complexity of manufacturing systems, and partly because of their dynamic and stochastic behaviour (Carrie, 1988; Kochhar, 1989; Law and Haider, 1989).

Choosing the right simulation package for the task is vital. Such packages are often very expensive, so finding the right one is cost effective not only in monetary terms but in the saving of time and effort that might otherwise be spent in adapting a less appropriate package at a later date. Choosing the right package is, however, not straightforward. The package should be capable of effectively modelling the two levels of detail described above: the conceptual, and the detailed design level. In this paper, a sample of the existing simulation packages are investigated with a view to examining their ability to model these different levels of detail. A brief overview of systems design is presented, and the manufacturing systems design (MSD) approach is described in further detail. Assessment criteria for modelling both the conceptual and detailed design levels are presented, and these are used to evaluate the packages used in this research. The types and characteristics of simulation packages available are described, and the ability of each of the three packages used in this research is analysed against the criteria for assessment. The paper concludes by suggesting a four-step framework that may better assist in the modelling of detail variability within MSD. An earlier version of this research has already been published (Eldabi and Paul, 1997). The next section describes systems design, as well as the MSD approach that is used to design a manufacturing system. It also offers reasons as to why simulation modelling is proving to be an increasingly popular tool in the design process.

2. MSD and CAMSD

There are two common approaches to systems design: top-down and bottom-up. Top-down begins by setting the objectives, then creating a system model that fits these, paying less attention to the current situation. A bottom-up approach is based on the existing system, producing a design that requires less capital investment. The MSD approach, introduced by Wu (1992a), is a combination of the top-down and bottom-up approaches. It initiates the project by analysing the current and the desired future positions. A set of objectives can be identified by analysing the desired future position under the constraint of current position analysis findings. The Computer Aided Manufacturing Systems Design (CAMSD) Research Group at Brunel University aims to develop a formal framework for manufacturing systems design (MSD) which will be implemented within a flexible IT environment. In this way, MSD engineers will be supported by a complete CAMSD solution using any chosen MSD methodology (CAMSD, 1994). The structure of the design process, as shown in Figure 1, can be
summarised as follows. The first two stages are analysis of situation and setting of objectives. These require analysis of the current state of the manufacturing organisation. They initiate an analysis of current markets and their future prospects. The following steps in the framework are design phases, which transform this operation from the current state to the desired state.

The next stage is conceptual modelling, that is, identifying the building blocks or the overall structure of the new system that will achieve the desired results. These blocks will be a combination of manufacturing functions and their relationships, together with necessary controlling functions. At this level, higher-level specifications must be developed for the function and data design. In addition, the long term production capacity to be achieved will also have been specified in terms of average static capacity levels and levels of variation which reflect the dynamic requirements (Schroeder, 1993; Tanner, 1985; Vanderspek, 1993). The completion of the conceptual design of the system creates a system model in terms of related functions. It contains a set of inter-related manufacturing functions each with a related list of products. There is also a hierarchy of control systems that will process key information associated with the effective performance of these functions. In summary, the detailed design phase takes the conceptual model of the selected manufacturing systems options and transforms this into a detailed specification that can be used later for implementation. Evaluation and decision phases take place after the conceptual modelling and the
detailed design. At this level the design team selects the best available business option with regard to their objectives.

MSD is a formal method or a systematic approach that is used to design a manufacturing system. The CAMSD objective is to produce a framework to support manufacturing engineers in the design of production operations. There are two main characteristics of such a framework. One is that it must be flexible enough to address the variety of situations encountered in manufacturing system design, while being systematic, pragmatic, and practical. The second is that it should be presented to engineers in a form that increases their motivation to use it in practising, focusing and guiding their activities (CAMSD, 1994; Fritz et al., 1994).

3. Simulation in manufacturing
During the last three decades there has been a dramatic increase in the use of simulation to design and optimise manufacturing and warehousing systems (Hollocks, 1992). There are three main reasons for the increase in use of simulation in manufacturing. Firstly, increasing competition as a result of greater emphasis on automation to increase productivity, quality, and reduce costs, has led to an increased complexity which can be analysed only by simulation. Secondly, there has been a large-scale reduction in the cost of computer hardware required to run the simulation models, in addition to the availability of advanced simulation software. Thirdly, the introduction of animation has resulted in a greater understanding of simulation by non-simulationists such as managers and manufacturing engineers. The types of simulation software tools are described in the following section. A representative sample of simulation tools is analysed, and these are assessed with a view to finding out how well they are able to model both the conceptual and the detailed design features described.

4. Simulation software: types and assessment
During the early days of the application of simulation techniques, many simulation models were created using high level programming languages such as FORTRAN and Pascal, or general purpose simulation languages such as GASP, GPSS, SIMSCRIPT, SLAM, and SIMULA (Kochhar, 1989). However, during the last two decades, many simulation software tools that require less or no programming effort and experience have become commercially available. Examples of these tools include SIMFACTORY II.5, ProModel for Windows, AutoMod II, and WITNESS. Law and Kelton (1991) present the features desired in simulation software, which can be summarised as follows:

- generating random numbers from the uniform probability distribution;
- generating random values from a specified probability distribution;
- determining the next event from the event list and passing control to the appropriate block of code;
- adding records to, or deleting records from, a list;
- advancing simulation time;
- detecting error conditions;
- collecting and analysing data; and
- reporting the results.
Simulation packages vary widely in their features and characteristics, and there are many methods for classifying them. This paper describes one of the methods of classification, where simulation software tools are classified into three categories:

1. General purpose simulation languages.
2. Data-driven simulators.

The choice of this classification is mainly based on the objectives of CAMSD, which are related to agility of the modelling tool and suitability to each level of detail. From the description above with regard to CAMSD it is clear the complexity and modelling speed are possibly the main factors to look for when searching of a package that caters for both the conceptual level and the detail level. For the purpose of our investigation we selected one package from each category, which are further described below. The packages chosen are supposed to be typical to its corresponding category.

**General purpose simulation languages**

A general-purpose simulation language (GPSSL) is a simulation package that is used for modelling different types of systems with different characteristics. A typical GPSSL usually compels the users to manually conduct the tasks specified in the previous section. There is usually extensive amount of low-level coding. The model is usually developed by coding from scratch. A typical GPSSL, although purposefully developed for simulation, has no specific orientation towards a typical type of models. In fact it can be used for any application area. However, Law and Haider (1989) state that some of these languages may have special features for manufacturing such as workstation and material handling modules. For example, AutoMod II is specifically directed towards material handling and manufacturing problems. Another example of such a package, SIMAN/CINEMA IV, has special material handling features, such as forklifts and conveyors (Pegden et al., 1990). In this research, we choose SIMAN/CINEMA IV to represent this category.

**Data-driven simulators**

Law and Haider (1989) define a data-driven simulator (DDS) as a computer package that allows the modeller to model systems with little or no programming. DDS can be considered as the other extreme of GPSSL. The fundamental difference between DDS and GPSSL is possibly attributed to the visual features and the speed of model development. Many data-driven simulators are domain-specific (Pidd, 1998). They are used to model systems with specific features (e.g., cellular manufacturing systems). There are simulators currently available for certain types of manufacturing, computer, and communication systems. Examples of simulators, which are dedicated to manufacturing simulation, are: SIMFACTORY II.5, WITNESS, ProModel for Windows, and Xcell +. A graphical user interface is a fundamental part of simulators, which is used for modelling as well as for running the model. Most of these simulators employ a network in their underlying concept (Pidd, 1998). Thus entities are assumed to flow through a network from node to node. At these nodes they may be delayed as they engage in activities with entities and resources placed on the nodes. Resources placed on the nodes may also be engaged in endogenous activities. For example, in a manufacturing application such entities can be machines that
occasionally fail. In this research, SIMFACTORY II.5 is selected to represent this type of software.

**Program generators**

Program generators are used as another way of making simulation more accessible to non-computer specialists. A program generator is a computer program with a user interface that enables users to generate another program (code). Unlike a compiler, which takes a source program written in a problem-oriented language and produces machine code, a program generator takes a system description and produces source code. This generated source code may then be compiled or interpreted to present a computable simulation model (Pidd, 1998). Examples of program generators are CAPS/ECSL, VS7, and DRAFT. Program generators are usually interactive and accept a description of a conceptual model such as an activity cycle diagram (ACD) (Carrie, 1988). The ACD is a graphical representation that depicts the cycle of each type of entity. Each life cycle is a closed cycle of alternating activities and queues (Paul and Balmer, 1993). Most program generators require definition of the model entities, activities, queues, attributes, and priorities. Thus the user starts modelling by drawing an ACD, and then describes the components of the diagram to the program generator. Generally, features of program generators lie between those of simulation languages and data-driven simulators. VS7 is taken in this research as a representative of this software category.

In order to investigate detail variability during model building and running of the simulation packages investigated as part of this research, assessment criteria were drawn up to assess the ability of the packages to model both the conceptual modelling level and detailed design level of MSD. The assessment criteria for the conceptual level and for the detailed modelling are presented here in Table I.

It should be noted that this research concentrates on the criteria for selecting manufacturing simulation packages that can model problems with different levels of detail such as the conceptual model level and the detailed design level of the MSD. The criteria outlined above may be added to other groups of selection criteria in order to select a suitable package. The reader is referred to (Hlupic and Paul, 1995; Hlupic and Paul, 1996; Hlupic, 1997) for other evaluation and selection criteria. This section has described the features of both conceptual modelling and detailed design. An evaluation of the results is described in the following section.

**5. Results and analysis**

A typical manufacturing system has been used for analysing the three packages in question. The system used is a multi-level manufacturing problem case study described by Wu (1992b). Each of the packages has been used for modelling the system following the MSD framework concept. Findings and results of modelling – that is in addition to observations about the modelling process itself – for each package were collected and matched with the previously identified modelling features for both levels of details. In this analysis, the higher the ability of a package to meet both groups of modelling features, the higher will be its ability to offer modelling variability. The ability of each package to match each of the modelling criteria is described according to three levels: absence of, or poor quality of matching; fair quality of matching; and excellent quality of matching. The judging process is closely aligned with the
Conceptual modelling features

The conceptual level design needs to be quick and simple. It contains the essential blocks of the system, but without the detail. It is used for high-level analysis and long-term planning.

Modelling requirements: quick and simple model building
At this stage the conceptual model is supposed to be as simple as possible, as it contains the major blocks of the system. For the design process to be carried out efficiently, modelling needs to be done quickly. Quick modelling is a feature of simulation modelling packages, and may be affected by factors such as the time required to learn the simulation package in use, the complexity of model building in such a package, compilation, and debugging.

Modelling requirements: running-speed
This feature is important because it gives a clearer idea about the system in the long term, which helps in strategic planning. The simulation package must be capable of running models over long periods of simulated time but in shorter real time.

Modelling requirements: low-level animation
Animation in simulation is generally used for model debugging, verification and validation, and analysis. At this stage of the MSD, low-level animation is suitable. Low-level animation, or level I animation (Johnson and Poorte, 1988), can be characterised as a two-dimensional animation which includes only colour changes as indicators for changes in the state of entities. Icons used in this type of animation are abstract or basic geometric shapes. Generally, this level of animation displays the logic of the model without including physical layout, such as the real positioning of cells.

Table I.
Assessment criteria for modelling conceptual model and detailed design levels

Detailed modelling features

The detailed design level is needed to decide the detailed layout of the plant by selecting and allocating the required equipment.

Modelling requirements: detailed model building
The main feature of a model at this level is that it should contain all the details about the system specifications. The model must be as precise as possible, including all details about cell components and transportation behaviour. The detailed model is an extension of the conceptual model in that details such as variation of machines number in each cells, number of transporters, loading times, and unloading times are added.

Modelling requirements: automatic batch run
The detailed design is generally concerned with the daily or weekly performance of the factory. For this reason, simulation run-length is supposed to be a short period, usually between two weeks to one month. As the model at this stage is supposed to be precise, a number of independent replications should be made. This is to ensure that the results are accurate by quantifying variation due to sampling fluctuations.

Modelling requirements: high-level animation
Animation is important for analysis as well as communication with other people, who may not be experts in simulation (such as managers, manufacturing engineers, or clients). Animation therefore needs to be made meaningful, and in such a way that anyone can make sense of it. Usually the type of animation used here is level II animation (Johnson and Poorte, 1988). Level II animation is characterised as a two-dimensional system with movements of objects. Icons and symbols in some way depict the real parts.

(continued)
<table>
<thead>
<tr>
<th>Conceptual modelling features</th>
<th>Detailed modelling features</th>
<th>Business process design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Desirable results for purposes of analysis: total output</strong></td>
<td><strong>Modelling requirements: manufacturing features</strong></td>
<td>29</td>
</tr>
<tr>
<td>Total output is desirable at this stage to measure the performance of the system as a whole. It is used to compare the different configurations of the conceptual model. That is, to examine which configuration produces the highest output where all configurations are assigned the same types of resources.</td>
<td>It is very important at the detailed design level to have manufacturing facilities within the package used. Such features enhance the process of modelling a manufacturing system in a shorter period of time. These features could be machines, transporters, AGV, and conveyors (Cheng, 1985).</td>
<td></td>
</tr>
<tr>
<td><strong>Desirable results for purposes of analysis: cell utilisation</strong></td>
<td><strong>Desirable results for analysis: total output</strong></td>
<td></td>
</tr>
<tr>
<td>This feature is needed to examine the individual performances of different cells. It could be used as a comparison factor between different configurations. For example, if total outputs are equal for every configuration, the analyst may refer to cell utilisation results to determine which configuration would produce the same output with lower utilisation.</td>
<td>Total output is required at this stage to compare the performances of different configurations of the same detailed model. For example, to examine the effect of variation in the numbers of machines and transporters on each configuration.</td>
<td></td>
</tr>
<tr>
<td><strong>Desirable results for purposes of analysis: parts life-time</strong></td>
<td><strong>Desirable results for analysis: detailed statistics</strong></td>
<td></td>
</tr>
<tr>
<td>This feature is considered a very important method for measuring the performance of the system, as it is used to measure the time spent by a part in the system from the arrival time until all its corresponding processes are finished. This gives a good view about the process routing and helps in comparing different configurations of the system.</td>
<td>This feature is needed to detect which cells have large rates of accumulation of parts. This helps in identifying bottlenecks in the system, and in deciding where to build storage points and which cells need more machines. In addition, this feature is required to add or delete one or more transporters based on the queuing statistics for these.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.

description of each criterion in Table I. For example, if the package does not provide flexibility to build sophisticated models of complex systems then a “poor” mark is given to it within the “detailed modelling” criterion. The reason for choosing such a simple methodology is due to the fact that using a complex methodology is not necessarily going to provide results that are significantly different, specially for this type of analysis. Table II now shows the summary results of the analysis of the three simulation packages with respect to both levels of detail.

Generalising from these results, it can be said that some of the simulation packages can be used for efficient modelling at the conceptual level, but are not sufficiently suitable for modelling greater detail. Other packages, however, might be considered to be too complex when used for modelling conceptual levels, but are more useful for modelling greater detail. An overview of these findings is best demonstrated in Table III.

Choosing the right simulation package for the task entails more than describing and assessing the important features separately, however. Central to the decision to use one particular package or another is the requirements of the modelling itself. As can be
<table>
<thead>
<tr>
<th>Conceptual modelling features</th>
<th>Detailed modelling features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick and simple model building</td>
<td>Absent/or poor</td>
</tr>
<tr>
<td>SIMAN/CINEMA IV</td>
<td>SIMAN/CINEMA IV</td>
</tr>
<tr>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>SIMFACTORY II.5</td>
<td>SIMFACTORY II.5</td>
</tr>
<tr>
<td>VS7</td>
<td>VS7</td>
</tr>
<tr>
<td>Running speed</td>
<td>Acceptance/Good</td>
</tr>
<tr>
<td>SIMAN/CINEMA IV</td>
<td>SIMAN/CINEMA IV</td>
</tr>
<tr>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>SIMFACTORY II.5</td>
<td>SIMFACTORY II.5</td>
</tr>
<tr>
<td>VS7</td>
<td>VS7</td>
</tr>
<tr>
<td>Low-level animation</td>
<td>High-level animation</td>
</tr>
<tr>
<td>SIMAN/CINEMA IV</td>
<td>SIMAN/CINEMA IV</td>
</tr>
<tr>
<td>Absent/or poor</td>
<td>Excellence</td>
</tr>
<tr>
<td>SIMFACTORY II.5</td>
<td>SIMFACTORY II.5</td>
</tr>
<tr>
<td>VS7</td>
<td>VS7</td>
</tr>
<tr>
<td>Total output</td>
<td>Manufacturing features</td>
</tr>
<tr>
<td>SIMAN/CINEMA IV</td>
<td>SIMAN/CINEMA IV</td>
</tr>
<tr>
<td>Fair</td>
<td>Excellent</td>
</tr>
<tr>
<td>SIMFACTORY II.5</td>
<td>SIMFACTORY II.5</td>
</tr>
<tr>
<td>VS7</td>
<td>VS7</td>
</tr>
<tr>
<td>Cell utilisation</td>
<td>Total output</td>
</tr>
<tr>
<td>SIMAN/CINEMA IV</td>
<td>SIMAN/CINEMA IV</td>
</tr>
<tr>
<td>Fair</td>
<td>Excellent</td>
</tr>
<tr>
<td>SIMFACTORY II.5</td>
<td>SIMFACTORY II.5</td>
</tr>
<tr>
<td>VS7</td>
<td>VS7</td>
</tr>
<tr>
<td>Parts life-time</td>
<td>Detailed statistics</td>
</tr>
<tr>
<td>SIMAN/CINEMA IV</td>
<td>SIMAN/CINEMA IV</td>
</tr>
<tr>
<td>Fair</td>
<td>Excellent</td>
</tr>
<tr>
<td>SIMFACTORY II.5</td>
<td>SIMFACTORY II.5</td>
</tr>
<tr>
<td>VS7</td>
<td>VS7</td>
</tr>
<tr>
<td>SIMAN/CINEMA IV</td>
<td>SIMFACTORY I1.5</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1 excellent</td>
<td>4 excellent</td>
</tr>
<tr>
<td>3 fair</td>
<td>1 fair</td>
</tr>
<tr>
<td>2 absent or poor</td>
<td>1 absent or poor</td>
</tr>
</tbody>
</table>

**Table III**

Distribution of qualities of matching with the conceptual and detailed design modeling features.
seen from the earlier sections in this paper, the task of building models with detail variability is made more complex in that there is a great deal of variety between the existing simulation packages that are currently available. In the next section we present useful guidelines for assisting in developing models with detail variability.

6. Proposed framework for modelling detail variability

This section focuses on the process of modelling the different levels of detail of the formal framework of manufacturing systems design (MSD). Conceptual models may be built by making effective use of the conceptual modelling features that are available in today’s simulation packages, and we present here how these may be flexibly extended to include detailed design features. This four-step method is described in Table IV.

This is an overview of how system components can be classified for flexible modelling of detail variability at the conceptual level. It shows the identification of the components and what the composition of their internal structures are. It then demonstrates the relationship between these interdependencies and how information at the detailed level can be entered into the simulation package. As we see it, this proposal is not developing an agile simulation package, in fact it is about the process of modelling itself. We see it as a way forward for coping with multiple details MSD framework. The underlying principle here is the way we conduct modelling rather than what tool to use. One way to support the proposed framework is the introduction of add-ins. An add-in usually supports certain aspects of the model. For example, an add-in for a vehicle that goes from one node to another whilst transporting goods. These add-ins are not usually provided with the package, but can be acquired once the users feel it would contribute to the quality of the model. Given our proposed framework, such add-ins could be inserted at the detailed level after the conceptual model has been agreed as acceptable by the stakeholders. Of course not all simulation packages have such features nor are all expected to, yet the bottom line of our stance is that to cope with the variable detail MSD framework, then the modelling framework proposed in this paper is one way to go ahead and add-ins is one idea that can support this proposal.

7. Conclusions

This paper investigates three simulation packages, each one representing a simulation software category. The investigation focuses on the ability of these packages to handle problems concerning business process design taking into account different levels of detail portrayed in the MSD framework – described in section 2. The objectives of this research were not only to investigate the existing simulation software environments that are able to offer variable detail modelling, but also to better understand the complex relationships between the various components that comprise the model. Findings from this research have suggested that there is currently no such simulation package that can meet all of the requirements suggested by CAMSD, although some of the packages meet some of the requirements at either the conceptual or the detailed level, but not at both. SIMAN/CINEMA IV, for example, is suitable for detailed modelling, whereas SIMFACTORY II.5 is more suitable for conceptual modelling.

One important finding from this research is that simulation packages tend to follow certain objectives. For example, DDS packages tend to be concentrating on developing quick and dirty models and the same applies of the other. We see the main reason for
Step 1 (conceptual modelling)  Identification and classification of the main blocks or entities of the system separately  The first major step in building such a model should be the identification of the building-blocks of the system, such as the different types of cells. When starting to identify the basic components of the conceptual model, the modeller must bear in mind that this model is to be extended into more detail later in a flexible manner without the need to create a new detailed simulation model from scratch. Therefore, the major components of a conceptual model might be classified as separate, preferably non-overlapping blocks or entities regardless of their internal structures and details. These will be expanded into more detail later.

Step 2 (conceptual modelling)  Assigning averages and assumptions of real data to the established blocks  At this stage, little detail is entered. After the identification of the main entities of the model, the second step is to assign the behaviour of each entity. Generally, the modeller, when developing the conceptual model, must avoid including any unnecessary details that may overcomplicate the conceptual model. On the other hand, missing out any other important components at this level will increase the problem of complication in the more detailed stage. If a simulation model of the conceptual level is built correctly, it will provide the required results and at the same time it will be a well-established base for detailed design. This can easily be extended with more details and complexity. At this stage the modeller may assign equal numbers for each entity or resource. For instance, assigning the same number of machines for each cell. Another example might be the assigning of equal speed of transportation between any two cells. This is to eliminate the effect of such details on the simulation results. Generally speaking, the model at this stage is not necessarily “valid” or typical of the real system.

Step 3 (detailed design)  Adding more detail (entities and activities)  These are the details needed to build the final model, and include essential factors such as physical layout. Here the new details within the boundaries of the blocks are entered; these have already been created at the conceptual level. That is, each block of the conceptual model is expanded separately from the rest of the model. Detailed data of a cell block could, for example, be the number of machines in the cell, the process duration of each machine, the rate of failure for each machine, and maintenance time. Sometimes it can be expanded into an internal network of activities. For example, in a “painting-cell”, block parts may be queued for cleaning, then after cleaning they are transferred to another queue for painting. Some details might be entered as interactions between different entities such as, physical positioning, distances, and directions between cells within the system.

Table IV. Four-step data classification framework (continued)
such specialisation is due to the strategy of the vendors or for marketing for specific domain. Some packages are produced for high-level managers who might not be particularly interested in detail, they only relate to strategic corporate decisions, whilst some other packages are produced for shop level management who might be taking day-to-day decisions, making them more interested in the detail of the process. The current technology does not allow for developing a package for modelling quick models and complex models in the same time. However, the introduction of plug-ins can be seen as one way ahead. With the emphasis today on component-based systems, further work is needed in order to look at not only different types of individual packages but also at their ability or otherwise to interact with others. Currently, there is the new trend of compatibility amongst different packages. Some simulation packages are compatible with graphical packages such as VISIO™. In other words, a conceptual model can be developed in a graphical package then auto-converted into a simulation package to add more detail. These are all alternative solutions, which could be explored in the future but also might become obsolete once a comprehensive package is developed that caters for all features together.

The problem of modelling variable detail depends for the most part on the methodology of modelling itself, that is, how the entities of the model are classified (Toncich, 1992). The framework described and introduced in this paper identifies a suitable way to perform entity classification with regard to the levels of detail of the MSD framework. This will enhance the performance of business process design as it offers a mechanism for incorporating different levels of detail. This research reports on work done in the area of manufacturing systems design, but is applicable in many other areas. It would be useful to broaden this to include, for example, delivery systems, where logistical behaviour needs to be effectively modelled. It may also be useful in banking, where the flow of information needs to be better understood. A third example is in the area of pharmaceuticals, where there is a need to take a more
long-term view of information when developing medicine. This research looked only at three individual packages. Further development in this area would serve to enhance the flexibility of simulation modelling packages in order to meet the increasing complexities of business process design with variable levels of detail.

References


CAMSD (1994), CAMSD Newsletter, Brunel University, London.


