Methodology for rapid identification and collection of input data in the simulation of manufacturing systems

T. Perera *, K. Liyanage

Modelling and Simulation Research Group, School of Engineering, Sheffield Hallam University, Sheffield S1 1WB, UK

Received 18 January 1999; received in revised form 1 June 1999

Abstract

Computer simulation is a well-established decision support tool in the manufacturing industry. The rapid development and deployment of simulation models however, are inhibited by factors such as inefficient data collection, lengthy model documentation, and poorly planned experimentation. Typically, more than one third of project time is spent on identification, collection, validation, and analysis of input data. Whilst most research work has been focused on statistical techniques for data analysis, less attention has been paid to the development of systematic approaches to input data gathering. This paper presents a methodology for rapid identification and collection of input data in batch manufacturing environments. A functional module library and a reference data model, both developed using the IDEF (Integrated computer aided manufacturing DEFINition) family of constructs, are the core elements of the methodology. The paper also identifies the major causes behind the inefficient collection of data. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Simulation; Input data modelling; IDEF

1. Introduction

Computer simulation is now seen as an integral tool in the design, planning, operation and restructuring of manufacturing systems [6,22]. Traditionally, simulation is used in capital intensive projects such as the design of new factory layouts. The availability of affordable and user-friendly software tools have improved the
usability of computer simulation it is therefore frequently used to address a wide variety of operational problems [19,3]. As a result, the ability to develop and deploy simulation models quickly and effectively is far more important than ever before. However, a number of factors such as inefficient data collection, lengthy model documentation, and poorly planned experimentation prevent frequent deployment of simulation models.

A serious limitation among the above factors is inefficient data collection. It is obvious that the development of simulation models is delayed when the right data is not available in the right format at the right time. On the basis of data collated from a number of industrial applications, Trybula [20] argues that in a typical model building exercise, up to 40% of the project time is required by data gathering and validation. In fact, it seems that the effort required to collect data has not significantly changed over the last decade.

In the 1980s limitations of simulation software often excluded complex features from the modelling process; consequently, projects required less data. However, data had to be collected by manual means. Advances in simulation software have enabled the modellers to build more complex models in the 1990s, requiring large volumes and variety of data. Thus, the effort required to gather and analyse data remain somewhat the same.

This paper presents a methodology based on IDEF family of tools for rapid identification and collection of input data. Batch manufacturing was chosen to demonstrate the use of the methodology.

2. Input data collection

Prior to the development of a methodology for rapid data identification and collection, it was necessary to identify the major causes of inefficient data collection. Information related to the collection of input data was obtained from three sources: literature, interviews with simulation practitioners, and a survey conducted at the 1997 WSC (Winter Simulation Conference). A number of authors have raised issues surrounding data collection [2,4,7–9,13,15,17,20]. Analysis of the collected data led to identification of the following seven major causes.

2.1. Incorrect problem definition

Generally, one or more symptoms, such as longer lead times and under-utilised resources, trigger the need for a systematic analysis. Before simulation is chosen as the preferred method of analysis, good understanding of the nature and scale of the problem(s) are required. Shannon [18] argues that in some simulation exercises, many millions of dollars are spent developing exotic solutions for the wrong problem. Simulation projects initiated with poor understanding have a higher risk of failure due to excessive time being invested in collecting inappropriate data [21].
2.2. Lack of clear objectives

One of the most important aspects of a simulation project is the clear definition of project objectives [5,16]. The lack of clear objectives may impact every aspect of a simulation project. For example, poor definition of objectives can affect the scope of the model, leading to inappropriate model details [12]. If the appropriate level of model details is not determined at the outset, the gathering of required data becomes difficult.

2.3. High system complexity

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

2.4. Higher level of model details

The level of model detail has clear implications for data collection. The level of detail depends on project objectives, data availability, creditability concerns, computer constraints, and the opinion of system 'experts' [5]. A higher level of model detail does not necessarily lead to higher accuracy and comprehension, but can lead to longer data collection [16]. A typical entity (data group) may have many attributes that naturally include some core attributes. These are regarded as essential data elements for modelling of the system. For example, process time is a core attribute. Collection of data related to core attributes is less time consuming and does generally not require much effort. As the model detail increases, more and more non-core attributes may be introduced. Collection of data for these non-core attributes may take longer time. For instance, more effort is required to collect data related to machine breakdowns.

2.5. Poor data availability

When model details are increased, it may be difficult to find quality data for new attributes [4,16]. The required data may simply not be available. Simulation models are also built for new manufacturing systems being planned. In such cases, the modeller may not be able to collect the required data, due to the unavailability of past operation [6,14,15]. Table 1 depicts how the participants of the 1997 Winter Simulation Conference perceived the effort required to collect a selection of data items.
2.6. Difficulty in identifying available data sources

Simulation models may need data from numerous sources. These sources can vary from simple manual systems to sophisticated computer based systems. These systems can often provide the required data. Model builders, however, find it difficult to identify reliable data sources due to:

- The existence of multiple data sources for the same data type. For example, processing times of parts may be found in both MRP II and process planning systems. Due to the lack of integration, these two sources may provide different values for the same type of data. This uncertainty in data may force the model builder to seek a third party’s opinion in order to identify a more accurate source of data.
- Indirect existence of data. For example, the data required to model machine breakdowns (i.e. mean time between failures and mean time for repairs) may not be directly available. The maintenance department or maintenance contractors may have the required data, in very crude form. Hence considerable time is needed to collect and analyse this data.

2.7. Limited data handling capability

Most simulation packages do not offer facilities to organise and manipulate data; proprietary data formats are often used. When the core data is altered, it may be necessary to regenerate the data for simulation models. In large-scale projects, this can be extremely time consuming. This problem has been partially addressed via interfacing to spreadsheets and external databases.

The participants of the 1997 WSC were asked to assess the impact of these factors on data collection (Table 2). Poor data availability was considered to be the major cause of long data collection time. It is evident from the above discussion that a more methodical approach is required to identify and collect required data. Perhaps, the best way is to develop a guiding system, which could take model builders through a series of tasks to ensure complete and efficient data collection. The essence of such

Table 1

<table>
<thead>
<tr>
<th>Effort Required to Collect the Different Categories of Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Effort</strong></td>
</tr>
<tr>
<td>High</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Low</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
a methodology is a complete data model of the system. Once the core data types are
identified using a reference data model, they can be linked to potential data sources.
Depending on the availability of data sources, the model builder can be directed to
collect the necessary data quickly.

3. Methodology for rapid identification and collection of data structure

Our methodology is developed around a library of functional models and a gene-
ric data model for batch manufacturing environments. After analysing the features
of batch manufacturing systems, we developed a comprehensive data model and a
series of functional models with different level of details. Functional modules are
linked to appropriate data groups, via a mapping table. Once the required data types
are identified, data related to them is stored in a database, which can be directly
linked to the simulation software. Hence, the main elements of the methodology
are shown in Fig. 1:
1. a functional model (IDEF0) library (defined in 3.1),
2. a reference data (IDEF1X) model (defined in 3.2),
3. a mapping table to match functional models with reference data model (defined
   in 3.3),
4. database.

3.1. Functional model library

The IDEF0 Functional modelling methods are designed to model the decisions,
actions and activities of the system [10]. IDEF0 allow the user to “tell the story”
of what is happening in the system. The methodology permits a system to be de-
scribed as complete a level of detail as desired. We have developed a series of stan-
dard IDEF0 Functional Modelling diagrams for the system elements in generic
batch manufacturing systems, which are often modelled in simulation. For example,
Fig. 2 illustrates one of our standard IDEF0 functional modelling diagrams that de-
scribes the required machine activities concerning the part processing.
Fig. 1. An integrated approach to data identification and collection.

Fig. 2. An example of IDEF0 functional modelling diagram.
3.2. Reference data model

The reference data model has been developed using the IDEF1X [1,11] data modelling language. Its purpose is to describe the integration among various components such as parts, resources, and the logic of a manufacturing system into a single cohesive system to describe a conceptual database implementation. The reference data models show the major entities (data groups) with their attributes and relationships. They currently contain 24 entities and over 75 attributes. This model was translated into a normalised relational database. For a relational DBMS, each entity in the reference model becomes a table and each attribute becomes a column. Primary and foreign keys are declared for each table, and referential integrity constraints are declared for each relationship. Fig. 2 describes some parts of the reference data model as an example.

Fig. 3(a): Entity relationships between part, machine, and operations are described in the reference model as follows. Machines can perform many operations, and operations can be performed on many machines. An operation relates to many parts and parts, can undergo many operations. In the Reference model, these “many-to-many” relationships can be resolved in three “one-to-many” relationships with the associative entity MACHINE_OPERATION. The figure shows that a MACHINE_OPERATION represents a three-way association among PART (Identified by Part-ID), MACHINE (Identified by Machine-ID), and MACHINE_OPERATION (Identified by Operation-No).

Fig. 3(b): An important part of the manufacturing system is the movement of materials from one point to another (material handling systems). Our reference data model supports the identification and collection of relevant material handling data necessary for the simulation model. This figure provides a brief description concerning the material handling system, i.e. Parts can be transported by many MH_DEVICES, and MH_DEVICES can transport many PARTs. In the Reference model, these many-to-many relationships can be resolved by two one-to-many relationships with the associative entity TRANSFER_OPERATION (identified by Beginning Stn, Ending Stn, and Part No). MH Device (the generic parent entity) is a Conveyor, a Free-path transporter, or a Guided-path transporter (the categories).

Fig. 3(c): A bill of material (BOM) structure can be represented by two entities PART and ASSEMBLY-STRUCTURE in the reference model. The entity PART has a dual relationship as a parent entity to the entity ASSEMBLY-STRUCTURE (Identified by Parent-component.Part-ID and Child-component.Part-ID).

3.3. Mapping table

The purpose of a mapping table is to integrate both the functional modelling diagrams and the reference data model so that the modeller can identify system activities, corresponding information, and data quickly. A sample of mapping table, shown in Table 3, maps the manufacturing activity functions for the part processing against the entity and attributes represented in the reference data model with the corresponding Arena simulation software program constructs. This mapping table can
be extended further to represent the modules of other simulation packages as well. The development of this mapping table to Arena and other simulation packages will assist novice modellers.

4. Use of the methodology

**Step I: Investigate the system.** At this stage the model builder is required to identify the characteristics and operations of the system under investigation. A variety of methods are available to the model builder: interviews with stakeholders, walk-
<table>
<thead>
<tr>
<th>Level of decomposition</th>
<th>Functional modelling element</th>
<th>Required data</th>
<th>Reference model path (corresponding RM entity)</th>
<th>Corresponding ARENA template</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-A0</td>
<td>Part</td>
<td>Part ID</td>
<td>PART CREATE</td>
<td>CREATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Part description</td>
<td>PART CREATE</td>
<td>CREATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Batch size</td>
<td>MACHINE_OPERATION CREATE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Max batches</td>
<td>MACHINE_OPERATION CREATE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inter arrival time</td>
<td>MACHINE_OPERATION CREATE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Machine</td>
<td>Machine ID</td>
<td>MACHINE RESOURCE/SEIZE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Machine description</td>
<td>MACHINE RESOURCE/SEIZE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MTBF</td>
<td>MACHINE_GROUP RESOURCE</td>
<td>RESOURCE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MTTR</td>
<td>MACHINE_GROUP RESOURCE</td>
<td>RESOURCE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input buffer capacity</td>
<td>MACHINE_GROUP RESOURCE OR STORAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output buffer capacity</td>
<td>MACHINE_GROUP RESOURCE OR STORAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operator</td>
<td>Operator ID</td>
<td>OPERATOR RESOURCE/SEIZE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operator description</td>
<td>OPERATOR RESOURCE/SEIZE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Efficiency</td>
<td>OPERATOR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Skills</td>
<td>OPERATOR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learning curve effect</td>
<td>OPERATOR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Part</td>
<td>Machine operation desc.</td>
<td>MACHINE_OPERATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Processing</td>
<td>Machine time</td>
<td>MACHINE_OPERATION DELAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operator responsiblities</td>
<td>MACHINE_OPERATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Schedule</td>
<td>Schedule name</td>
<td>SCHEDULE RESOURCE-&gt;Schedule block</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Schedule description</td>
<td>SCHEDULE RESOURCE-&gt;Schedule block</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Schedule Time</td>
<td>SCHEDULE RESOURCE-&gt;Schedule block</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Duration</td>
<td>SCHEDULE RESOURCE-&gt;Schedule block</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Machine available time</td>
<td>MACHINE_SHEDULE RESOURCE-&gt;Schedule block</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Machine unavailable time</td>
<td>MACHINE_SHEDULE RESOURCE-&gt;Schedule block</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operator available time</td>
<td>OPERATOR_SHEDULE RESOURCE-&gt;Schedule block</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operator unavailable time</td>
<td>OPERATOR_SHEDULE RESOURCE-&gt;Schedule block</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instruction</td>
<td>Priority rule</td>
<td>MACHINE_OPERATION SEIZE</td>
<td></td>
</tr>
</tbody>
</table>
through of the system, and use of operating manuals etc. The primary objective is to identify the operating rules of the system. At Step II, these operating rules are translated into a series of IDEF diagrams using pre-built IDEF logic modules.

**Step II: Build functional model of the system using pre-built IDEF constructs.** The main objective of this step is to develop a complete functional model of the system under investigation using pre-built IDEF logic modules. A library of modules related to batch manufacturing systems is available to the model builder to create the reference model. Generally, these modules are assembled in hierarchical fashion so that more details can be shown at lower levels. The resulting functional model forms the basis for identifying the data requirements.

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

**Step III: Generate the required entity model using the mapping table.** The mapping table assists the modeller to identify the appropriate entities and their attributes. The range of attributes required for a given entity depends on the level of decomposition defined in the functional model developed in Stage II. The end result of this step is a customised entity model for the system under investigation. Most commercial packages available for IDEF modelling can automatically translate an entity model into a relational database. For instance; the entity model can be converted into a Microsoft Access Database.

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

5. Advantages and limitations

Our methodology has been successfully used in a number of industrial situations. During the development stage, data from different manufacturing companies were used to enhance the reference data model and the functional module library.

The main purpose of this methodology is to accelerate data identification and collection. It provides a systematic approach and moreover collected data is conveniently stored in a database that can be directly linked to simulation models. This approach can also support other stages of a simulation project; for instance, the need for detailed model documentation can be reduced via the use of standard modules from the functional library. Moreover, it is possible to partially automate the validation and verification of input data through boundary and input mask definitions.
Recent advances in simulation software, particularly integration via VBA (Visual Basic Application) makes it possible to create the entire simulation model automatically.

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

6. Conclusions

The data identification and collection are time consuming activities in simulation projects. The rapid development and deployment of simulation models are often constrained by problems input data gathering. The proposed methodology provides a tool kit to accelerate the collection of data. It utilises a data reference model, a functional module library, and a mapping table to create a database of required data. This methodology has been applied in real industrial situations and the results are promising.

Although batch manufacturing was used to demonstrate the methodology, it can be easily adopted for other areas by re-building the functional model library, the reference data model, and the mapping table. At present, a customised tool-kit is being built for the aerospace industry.

References