Methodologies for Data Collection and Model Documentation in Computer Simulation

K.N.H.P. Liyanage

Faculty of Engineering Technology, The Open University of Sri Lanka, Sri Lanka

Abstract. In recent years, computer simulation has become a mainstream decision support tool in an industry. In order to maximise the benefits of using simulation within businesses, simulation models should be designed, developed and deployed in a shorter time span. A number of factors, such as excessive model details, inefficient data collection, lengthy model documentation and poorly planned experiments, increase the overall lead-time of simulation projects. Among these factors, input data modeling and model documentation are seen as major obstacles. Input data identification, collection, validation and analysis typically take more than one-third of project time. This paper presents an IDEF (Integrated computer-aided manufacturing DEFinition) based approach to accelerate identification and collection of input data. A functional module library and a reference data model, both developed using the IDEF family of constructs, are the core elements of the methodology. In addition, this paper also intends to give a methodological approach that helps and motivates the project team to document simulation projects.

Keywords: Computer Simulation, Input data modelling, Model representation and documentation

1. Introduction

Simulation is more than building a model; it is a managing process, working with people and handling large amount of information (Piyasena et al, 2004). Simulation model development process typically consists of a number of different procedural steps, which are executed sequentially and/or iteratively. The process begins with analyzing of observations in real world system, collecting data and then formulating the logical system model for the development of a computer model that allows experimentation with. The method of conducting a simulation project is presented in different ways by different authors, but the conceptual approach remains the same in all procedures (Piyasena, 2004). Law (2003) presented a seven-step approach for conducting a successful simulation study.

The activities that take place in each step are shown in figure 1.

However, a number of factors such as inefficient data collection, lengthy model documentation and poorly planned experimentation prevent frequent deployment of simulation models. Most simulation practitioners argue that data collection takes considerably long time. It is obvious that the development of simulation models is delayed when the right information is not available in the right format at the right time. On the basis of research investigating a number of industrial applications, Trybula (1994) argues that, in a typical model building exercise, data gathering and validation can take up to 40 per cent of the project time.

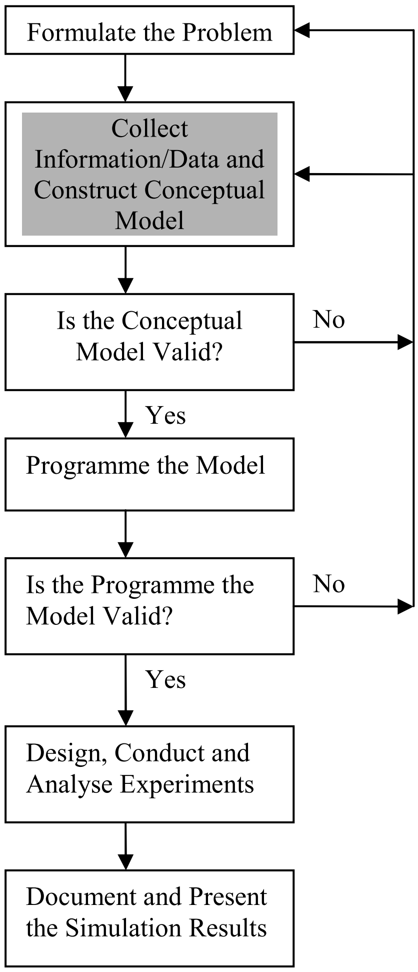


Figure 1. A seven-step approach for conducting a successful simulation study (Law, 2003)

Similarly conceptual modelling is an important part of the successful simulation process, because it influences the subsequence phases of the process. The formulation of the conceptual model is where the modeler decides what system elements to include in the model and to what level of detail they should be represented. Model documentation, mostly considered as model representation, is the process by which the conceptual model is translated into a form that can be communicated to another person and then translated into a form executable by a computer. In such an environment, proper communication through a well-documented procedure avoids early misunderstanding, minimizes project delays, and assists to ensure the validity and success of the project. Therefore, it appears that the sooner the documentation is dealt with, the faster the simulation project development would be (Piyasena et al, 2004). Hence, the documentation should be an integral part of the simulation process.

This paper intends to present a methodology for rapid collection of input data and procedures that help to project documentation.

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 relating to the collection of input data was obtained from three sources: literature, interviews with simulation practitioners and a survey conducted at the Winter Simulation Conference. Markowitz (1981), Hatami (1990), Dietz (1992), Trybula (1994), Lung (1995), Robinson and Bhatia (1995), Lehtonen and Seppala (1997), Oakshott (1997) and Rohrer and Banks (1998) are a selection of authors who raise issues surrounding data collection. Analysis of the collated data led to identification of seven major causes (see Figure 2).

3. Methodology for rapid identification and collection of data structure

This methodology is developed around a library of functional models and a generic data model for batch manufacturing environments. After analysing the features of batch manufacturing systems, a comprehensive data model and a series of functional models with different levels of details were developed. Functional modules are linked to appropriate data groups via a mapping table. Once the required data types are identified, data relating to them are stored in a database, which can be directly linked to simulation software. Hence, the main constitutes of the methodology are (Figure 3);

- a functional model (IDEF0) library ;
- a reference data (IDEF1X) model;
- a mapping table to match functional models with reference data model;
- a database.

Functional model library

The IDEF0 functional modelling methods are designed to model the decisions, actions and activities of the system (Mayer, 1992a). IDEF0 allows the user to ``tell the story'' of what is happening in the system. The methodology permits a system to be described in as complete a level of detail as desired. A series of standard IDEF0 functional modelling diagrams for the system elements in generic batch manufacturing system were developed. For example, Figure 4 illustrates one of the standard IDEF0 functional modelling diagrams, which describe the required machine activities concerning the part processing.

Reference data model

The reference data model has been developed using IDEF1X (Bruce, 1992; Mayer, 1992b) data modelling language. The purpose of this reference data model is to describe integration between various components such as parts, resources and 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. This model was translated into a normalised relational database. In this database, 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. Figure 5 describes some parts of the reference data model as an example.

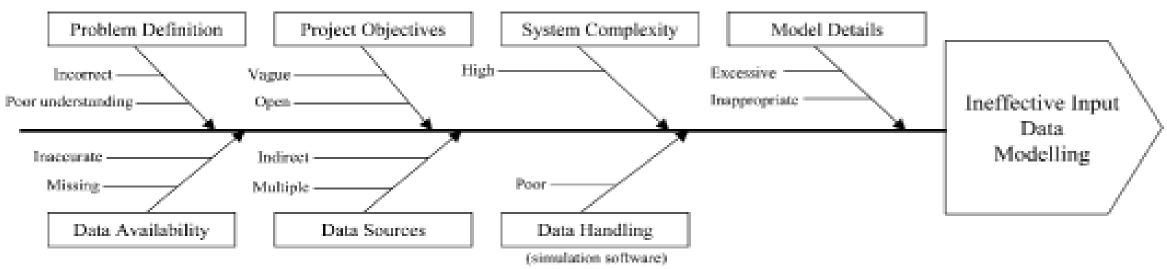


Figure 2. Causes for long data collection and wrong input data

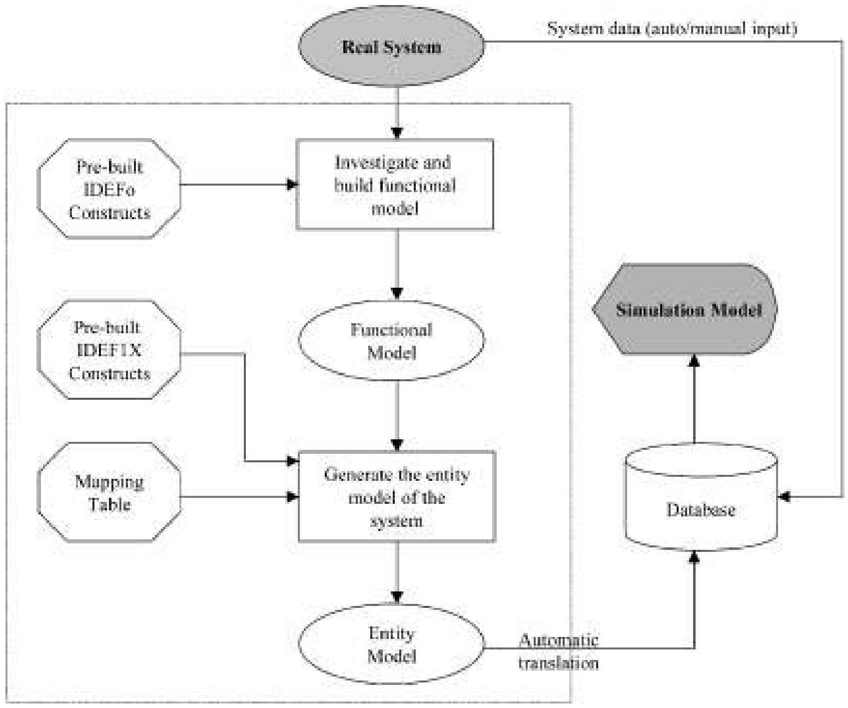


Figure 3. IDEF based approach

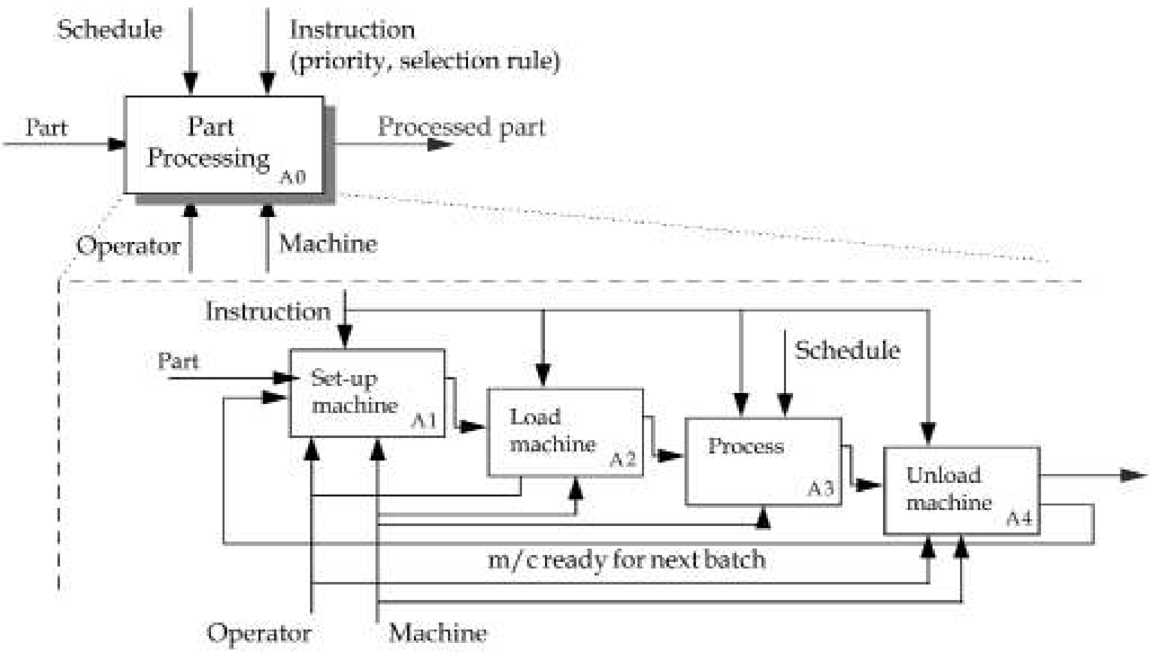
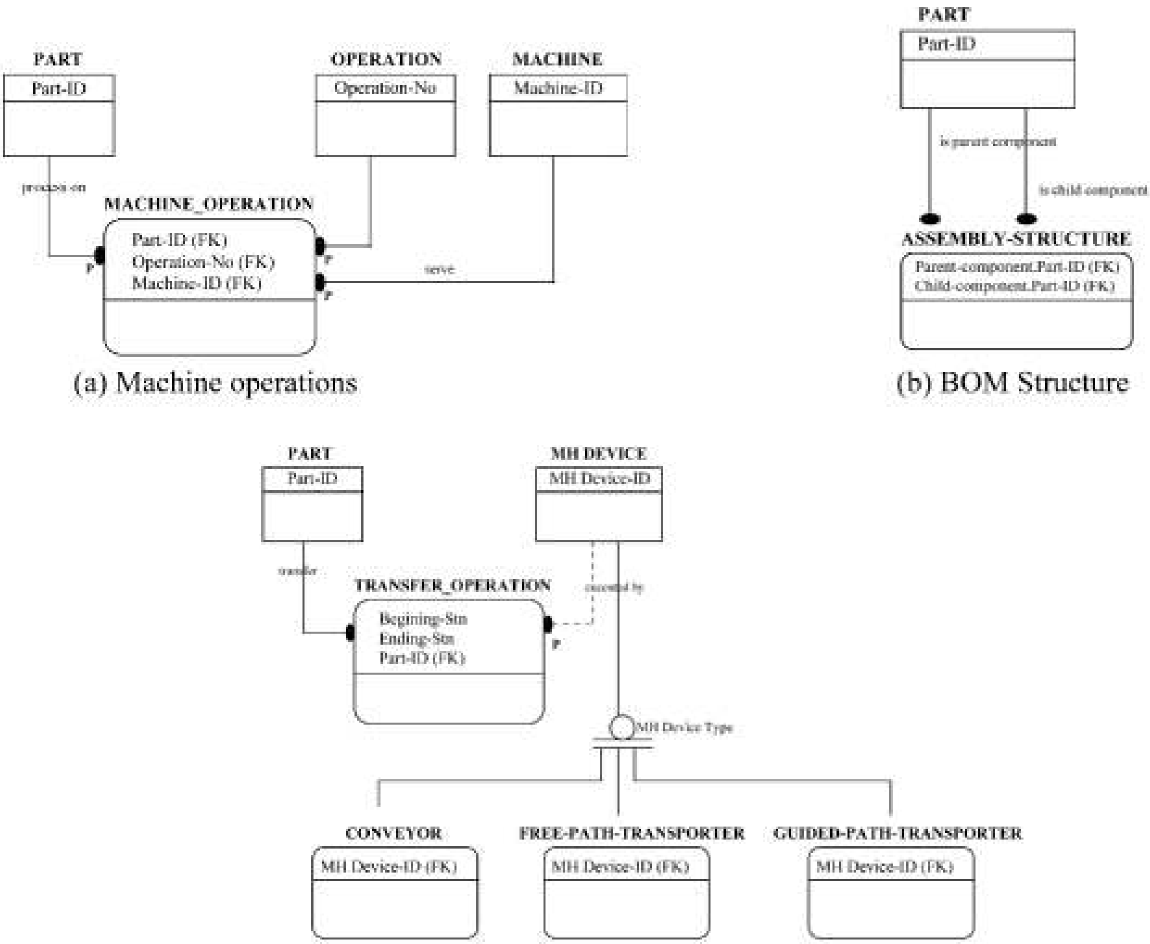


Figure 4. One of IDEF0 functional modules



(a) Machine operations

(b) BOM Structure

(c) Material movement operations

Figure 5. Parts of the IDEF1X reference data model

Figure 5(a): Entity relationships between part, machine and operations are described on 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 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).

Figure 5(b): A bill of material (BOM) structure can be represented by two entities PART and ASSEMBLY-STRUCTURE on 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).

Figure 5(c): An important part of the manufacturing system is the movement of materials from one point to another (i.e. 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 in two one-to-many relationships with 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).

Mapping table

The purpose of the 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 mapping table maps the manufacturing activity functions for the part processing against the entity and attributes represented in the reference data model with 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 such as interviews with stakeholders, walk-through the system and use of operating manuals are available to the model builder. 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 at this point is to develop a complete functional model of the system under investigation using pre-built IDEF logic modules. A library of modules relating 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 experimental factors, 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 is dependent 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 into data sources via the standard protocols such as ODBC.

5. Procedures for model representation and documentation

Particularly in model representation (or model documentation), the typical practice is to produce flowcharts or block diagrams, which are biased towards the simulation software being used, to represent some aspect of logics (Schormann and Perera, 1999).

As described in above, Pre-built IDEF0 functional model library can provide a systematic approach for model representation and documentation in manufacturing applications. These modules describe diagrammatic methods for conceptual modelling in discrete event simulation. It helps in structuring the ideas about the problems tackled in the projects, simplifies the task of understanding the logical design of a model, encourages high-level thinking and facilitates communication tool between developers and other parties.

There are alternative model representative techniques, approx. 20, that can be found in literature. Among these techniques IDEF, Petri Net, Unified Modelling Language (UML), Activity Cycle Diagrams (ACD) and Flowcharts are considered as potential model representation techniques for documentation (Piyasena, 2004). In this methodology, IDEF techniques have been used successfully. However, different techniques offer different types of representation with different advantages and disadvantages. Therefore, functional model library can be constructed using any technique, which is biased to simulation software being used, and it should be available in common graphical software platform. For example, instead of IDEF, flowcharts can be used to construct function model library. Visio, a leading desktop drawing product, supplies the common graphics platform for representing business process models with flowcharts; that document and organize complex ideas, processes, and systems. It is much popular in documentation of the software development. Through its open design based on customizable stencils, Visio readily fulfills the need to present different model representations from a common data repository. Visio's open architecture via ActiveX and VBA (Visual Basic Application) enables the overall modeling tool to be customized to fit changing needs. Arena Simulation software can capture process logic of a system which are drawn in the Visio, i.e process logic can be transferred from Visio drawings.

6. Conclusion

The main purpose of this methodology is to accelerate data identification and collection. In addition, it provides a methodological approach that helps and motivates the project team to document simulation projects. This approach can also support other stages of a simulation project. Moreover, it is possible to partially automate the validation and verification of input data through boundary and input mask definitions.

With recent advanced integration features in simulation software, in certain situations, it is possible to create the entire simulation model automatically. Nearly all the simulation software packages provide integration interface with other software for easy data transfer. For example, Arena provides ODBC, OLE, VBA and DXF interfaces; ProModel offers

OLE interface. Communication between the simulation software packages and the other software is generally accomplished through Dynamic Data Exchange (DDE), which most of the simulation software facilitate. In such an environment, it is possible to link databases (reference model) and process logic of a system (model representation) into simulation software.

7. References

Bruce, A.T. (1992), *Designing Quality Databases with IDEF1X Information Models*, Dorset House Publishing, New York, NY.

Dietz, M. (1992), ``Outline of a successful simulation project'', *Industrial Engineering*, November, pp. 50-3.

Hatami, S. (1990), ``Data requirements for analysis of manufacturing systems using computer simulation'', *Proceedings of the 1990 Winter Simulation Conference*, New Orleans, LA, pp.632-35.

Law, A.M. (2003), How to conduct a successful simulation study, , *Proceedings of the 2003 Winter Simulation Conference*, pp. 66-70.

Lehtonen, J. and Seppala, U. (1997), ``A methdology for data gathering and analysis in a logistics simulation project'', *Integrated Manufacturing Systems*, Vol. 8 No. 6, pp. 351-8.

Lung, A. (1995), ``Develop a consistent front-end approach to aid manufacturing simulation modelling'', *3rd International Conference on Manufacturing Technology*, pp. 455-9.

Markowitz, H.M. (1981), ``Barriers to the practical use of simulation analysis'', *Proceedings of the 1981 Winter Simulation Conference*, pp. 3-9.

Mayer, R.J. (1992a), *IDEF0 Functional Modelling*, Knowledge Base Systems, Inc., TX.

Mayer, R.J. (1992b), *IDEF1X Data Modelling*, Knowledge Base Systems, Inc., TX.

Oakshott, L. (1997), *Business Modelling and Simulation*, Pitman Publishing, London.

Piyasena S M (2004), Model representation and documentation in computer simulation, PhD Thesis (Un publised), Sheffield Hallam University, UK.

Piyasena S M., Perera TDS and Liyanage KNHP (2004), Task -Oriented Approach to Documentation in Computer Simulation, 2nd International Conference on Manufacturing Research, UK.

Robinson, S. and Bhatia, V. (1995), ``Secrets of successful simulation projects'', *Proceedings of the 1995 Winter Simulation Conference*, pp. 61-7. [193]

Rohrer, M. and Banks, J. (1998), ``Requires skills of a simulation analyst'', *IIE Solutions*, May, pp. 20-3.

Schormann F and Perera T (1998), Model representation methods in simulation of manufacturing systems, International conference on Simulation, IEE publication, University of York, UK, pp 292-96.

Trybula, W.J. (1994), ``Building simulation models without data'', *Proceedings of the IEEE International Conference of Systems, Man and Cybernetics*, pp. 209-14.