ADAPTABLE SIMULATION MODELS FOR MANUFACTURING

Jeffrey W. Herrmann[1], **Edward Lin**[1], **Bala Ram**[2], **Sanjiv Sarin**[2]
1: Institute for Systems Research, University of Maryland, College Park;
2: Department of Industrial Engineering, North Carolina A&T State University

ABSTRACT: Discrete-event simulation is a powerful tool for solving many problems, especially in manufacturing. Formulating the problem, building the simulation model, running the model, and analyzing the output are the basic steps in a simulation study. Because building a simulation model can be a difficult and time-consuming task, it will be useful if a decision-maker could reuse a simulation model if possible and change it to solve a different problem or evaluate another option. Thus, it is desirable to have adaptable simulation models that are easy to change with little or no programming effort. Using simulation models in real-time scheduling and operational settings also requires adaptable simulation models that can represent the changing shop floor. Also, as a manufacturing system progresses from a concept to a detailed design, and to an installed and operating facility, the simulation model of the system must change. Adaptable simulation models will reduce the time, effort, and cost of using simulation in these types of scenarios. This paper reviews the concepts of adaptability, suggests some measures for adaptable simulation models, and discusses factors that affect adaptability.

1. INTRODUCTION

Discrete-event simulation is a powerful tool for solving many problems, especially in manufacturing. Formulating the problem, building the simulation model, running the model, and analyzing the output are the basic steps in a simulation study. Because building a simulation model can be a difficult and time-consuming task, a decision-maker will seek to reuse a simulation model if possible and change it to solve a different problem or evaluate another option. Thus, it is desirable to have adaptable simulation models that are easy to change. Adaptable simulation models will reduce the time, effort, and cost of using simulation.

Several drivers are changing the construction and use of discrete-event simulation models in manufacturing enterprises. In global manufacturing enterprises, engineers must support more and more dynamically changing global markets and evolving paradigms, organizations and technologies for the next generation of production systems [1]. Simulation models are an important tool to these engineers [2]. In multinational corporations, information technology makes providing engineering services (such as in manufacturing engineering) from a remote location feasible and more common [3].

Simulation applications are more pervasive: Simulation is now being used for recurring operational tasks such as scheduling and production planning, beyond the traditional design and analysis tasks [4]. The availability of software packages and the computational power of personal computers have made it possible to bring computer simulation to the non-specialist [5].

The software reuse revolution also has had an impact. Object-oriented programming has made it easier to develop reusable software. Due to the large amount of effort required to develop simulation models, it would be worthwhile to build simulation models that are reusable. Some recent efforts are reported in [6] and [7].

Manufacturing corporations need to build and maintain discrete-event simulation models that represent their manufacturing operations. Over time, the manufacturing facility, the products, and the processes change. Modifying simulation models is often very difficult, however. Simulation models require a large amount of data. Small changes in the manufacturing environment can produce many different (though related) changes to the data input for the simulation model. Some examples of changes that are likely to occur are: (a) the answers needed from the simulation, (b) the products that are being made on the shop floor, (c) new production processes or characteristics of the current production processes, and (d) changes to the plant layout.

In addition, many manufacturing corporations use simulation models to evaluate the impact of moving a manufacturing facility to another location. When comparing a number of locations across the country or around the world, the analyst will have to modify the simulation model repeatedly to incorporate information about the specific location. This can be a very time-consuming effort.

Finally, the process of designing a manufacturing system requires changes to the simulation model. As a manufacturing system progresses from a concept to a detailed design to an installed and operating facility, the simulation model of the system must change. Typical changes include equipment selection and location, control rules and operating procedures for equipment and material handling systems, arriving material and customer order characteristics, and operating hours. [8]

Consider the example of processing time, a key piece of information in a manufacturing simulation model. Existing models require values for the processing times for all of the operations of each and every product. In fact, each value is a function of many inter-related inputs: product attributes like dimensions and material, machine capabilities like speed, and process parameters like temperature or pressure. A change in any input will change the processing time. Moreover, some inputs affect a large number of processing times. Changing a machine's capability, for instance, may change the processing time for all of the operations it performs. Current simulation methodology does not recognize the correlation between such data.

Another key problem with updating simulation models is maintaining the correspondence between the routing (process plan) of a product and the equipment that can perform the operations in the process plan. Typically, in a simulation model, a routing for a product specifies a given sequence of manufacturing workstations or machines. If some workstations or machines are replaced, then someone must manually update all of the routings that required those machines. Because this maintenance task is so time-consuming, it may never be performed. Instead, the inconsistencies are fixed only as they are found.

The remainder of this paper is organized as follows: Section 2 discusses past work on simulation and software reuse. Section 3 offers a definition for adaptability of simulation models and discusses factors that affect adaptability. Section 4 suggests possible measures for adaptability in simulation models. Section 5 describes case studies that we are developing. Section 6 concludes the paper.

2. RELATED WORK

Discrete-event simulation is a tool for modeling a system. Traditionally, simulation is used to describe and analyze the behavior of a system, to answer questions about proposed changes to the system, and to help one design a system. Simulation is often used for modeling and designing manufacturing systems like factories, flexible manufacturing systems, assembly lines, warehouses, and supply chains. Many other applications exist, including hospitals, military operations, traffic, airports, computer systems, and telecommunication networks. This paper focuses on manufacturing systems applications. For more information about simulation and its applications, see, for instance, the books by Banks and Law & Kelton. [9, 10]

Because simulation involves many calculations, computer programs are commonly used to create and run simulation models. There are a large number of simulation software packages available. [11] Modern simulation software packages vary, but a typical package includes the following components: a user interface, simulation models, a simulation engine, and output files. A user builds and changes a simulation model though the user interface. The user interface might be a text editor for manipulating statements in a simulation language or a spreadsheet for entering data, or a diagram that displays the resources in the system. The user interface allows the user to save these models for later use (and reuse). The user interface allows the user to specify parameters for the simulation runs and to start the simulation engine. Finally, the user interface provides methods for viewing and summarizing the output from one or more simulation runs.

The simulation model contains and organizes the data that describe the manufacturing system. This includes the resources (machines and people), the products, and the operations that each product requires. As with any other modeling effort, the user must specify the model's boundaries so that the simulation study can provide the necessary answers without requiring unnecessary detail, which increases the time and effort involved.

The simulation engine is the program that reads the data in the simulation model and executes one or more runs. In each run, the engine determines which events occur and when they occur. As the run progresses, the engine collects and output statistics about the system. In addition, it may output a complete list of all events.

In addition to the traditional applications like system design and performance estimation, real-time shop floor scheduling software systems are using simulation to create schedules. [12, 13] This software gathers data from a manufacturing execution system that tracks the progress of work in the shop. Then it updates a simulation model and uses that to create multiple schedules and select the best one.

Software developers have recognized the critical issue of software reuse and have developed methods for making software adaptable and easy to change. Gapers Jones lists five important subtopics under the general heading of "reusability": reusable data, reusable architecture, reusable designs, reusable programs and common systems, and reusable modules. [14, 15, 16] Goldberg and Rubin have a more object-oriented focus. [17] They suggest that most reuse falls into one of five categories: algorithms reuse, reuse of classes and instances, reuse of application frameworks, reuse of full applications, reuse of interface specifications.

Methods for developing reusable software have been proposed. Codenie *et al.* developed reuse contract methodology for managing reuse and evolution in software development. [18] This method addresses the incremental and iterative development of reusable software components and models and how to build highly customized applications based on these components and models. Lieberherr introduced the Demeter method for developing adaptive object-oriented software. [19] The Demeter method specifies the class dictionaries for defining the structure of objects and propagation patterns for implementing the behavior of the objects. This software is called adaptive since it adjusts automatically to a large number of context changes.

Several software reuse metrics have been developed. A reuse metric defines a way of measuring some attribute of developing software with reusable assets. Lim identifies a framework for reuse metrics that contain quality, productivity, and time-to-market metrics; reuse economic metrics; reuse library metrics; reuse process metrics; reuse product metrics; and reuse asset metrics. [16] The Goal-Question-Metric Paradigm and the Dashboard of Metrics are also described. Byrnes developed a model or production function for the comparison of adaptable software. [20]

3. ADAPTABLE SIMULATION MODELS

In the general context of decision-making, Mendelbaum provides the following definitions of flexibility: [21]

"Flexibility is the ability to respond effectively to changing circumstances."

"Action Flexibility is the capacity for taking new action to meet new circumstances." For example, planning without knowing the future as in plant expansion.

"State Flexibility is the capacity to continue functioning effectively despite change." For example, built-in robustness, absorbency, and tolerance to change.

In the context of manufacturing systems, there is a wealth of literature that deals with defining and measuring the flexibility of Flexible Manufacturing Systems. Buzacott states that the definitions of flexibility, action flexibility, and state flexibility apply well to the manufacturing systems environment. [22] In the manufacturing systems context, state flexibility refers to the ability to adjust or adapt to changes in the manufacturing environment automatically, while action flexibility requires manual intervention for adjustment.

Further, Buzacott has defined the following two approaches to addressing flexibility in manufacturing systems.

"Job Flexibility is the ability of the system to cope with changes in the jobs to be processed by the system." For example, being able to process a variety of jobs.

"Machine Flexibility is the ability of the system to cope with changes/disturbances at machines/workstations." For example, the capability of a machine to do a large variety of operations.

The above definitions of flexibility in the context of manufacturing systems are relevant to "adaptable simulation models" specifically for manufacturing systems. Adaptable simulation models, in our view, have the ability to handle the following types of changes:

- Requirements changes or changes in the answers to be provided by the simulation model.
- Internal and external changes in the production environment. Internal changes are changes in the process and material handling equipment and the interconnections among them. External

changes are changes to the products to be made and the production quantities, for example, which are not changes in the physical manufacturing system.

- Updated data provided by related information systems such as process planning and shop floor control.

4. ADAPTABILITY MEASURES

Adaptability measures the ease of changing a simulation model. The types of changes include those that Section 1 describes: changes to the real system that the model must incorporate, more detailed specification of the model, and changes to the questions being answered. Because so many types of changes exist, adaptability cannot be measured independently of the change being made.

We propose the following method for measuring a simulation model's adaptability. Given an existing simulation model M0 and a set of changes that need to be made, let M1 be the new simulation model that incorporates those changes. Let E(M1 - M0) be the effort required to create M1 by changing M0. Let E(M1) be the effort required to build M1 completely. Then the adaptability index of M0 with respect to M1 is A(M0, M1), and A(M0, M1) = E(M1 - M0) / E(M1). Thus a simulation model that is easy to change will have a index near zero. A simulation model that is hard to change will have a higher index, near one.

Effort can be a measure of time or cost. Since simulation models are data structures, we propose to measure effort by counting the number of data values that must be added, deleted, or changed. For simulation models that are sets of statements, we will measure effort as the number of statements that must be added, deleted, or changed.

Adaptability is a function of the particular model and the changes being modeled. Clearly, the simulation software package affects adaptability: the user interface, the modeling paradigm, and the data structures influence how many changes are necessary. Our research goal is to develop a simulation package and modeling guidelines that improve adaptability.

5. CASE STUDIES

This section describes an example that illustrates some concepts related to adaptability.

Manufacturing organizations and researchers have spent much effort in reducing manufacturing cycle times by improving manufacturing planning and control systems and developing more sophisticated scheduling procedures, and these efforts have shown success. However, it is clear that the product design, which requires a specific set of manufacturing operations, has a huge impact on the manufacturing cycle time. Product development teams need methods that can estimate the manufacturing cycle time of a given product design. If the predicted manufacturing cycle time is too large, the team can reduce the time by redesigning the product or modifying the production system. Estimating the manufacturing cycle time early in the product development process helps reduce the total product development time (and time-to-market) by avoiding redesigns later in the process. Thus, the product development team should include this activity in their concurrent engineering approach as they address other life cycle concerns, including testing, service, and disposal.

Adaptable simulation models are important to this activity. Given a simulation model that describes the factory operations when the new product will begin production, the product development team can modify the model by adding information about the new product, its process plan, estimated processing times, and the expected workload that the new product will create. The model can help the team estimate the new product's manufacturing cycle time and the increased cycle time of other products. In addition, the engineers can modify the model to evaluate the impact of adding more resources or making other changes to the factory that could accompany the new product's introduction.

We are currently looking at three case studies to build simulation models and provide a measure of adaptability for each of them.

In the first study we modeled the fictional but realistic Camile Motor Works (CMW) factory, described in a comprehensive planning and control reference case study. [23] The case clearly documents the factory's resources, products, and processes. The factory includes many types of manufacturing processes used to make a line of scale model automobiles. The factory displays many characteristics observed in actual factories. For the simulation study, we used the simulation software package Factory Explorer. Factory Explorer uses Microsoft Excel as the user interface. The user enters data about the products, the processes, and the resources in multiple Excel spreadsheets that follow prespecified formats. The simulation engine reads these worksheets and runs the simulation. We created an initial model of the CMW factory and two of its three product lines. Then we added a third product line. The effort required to modify the model was 36% of the effort required to create a completely new model. In the second case study we are modeling the CMW factory using ProModel. Thus we be able to determine how the simulation software affects the model's adaptability. The third case study addresses a manufacturing operation that makes electronics modules for automobiles; this is being done using Arena and seeks to study the building and maintaining of a simulation model for a functioning plant.

6. SUMMARY AND CONCLUSIONS

Building a simulation model can be a difficult and time-consuming task. Therefore, a decision-maker will seek to reuse a simulation model if possible and change it to solve a different problem or evaluate another option. Thus, it is desirable to have adaptable simulation models that are easy to change. Using simulation models in real-time scheduling and operational settings also requires adaptable simulation models that can represent the changing shop floor. Additionally, as a manufacturing system progresses from a concept to a detailed design to an installed and operating facility, the simulation model of the system must change. Adaptable simulation models will reduce the time, effort, and cost of using simulation in these types of scenarios. This paper has reviewed the concepts of adaptability and suggested a method to measure a simulation model's adaptability. The proposed adaptability index compares the effort needed to change the model to the effort required to build a completely new model. For illustration, this measure was applied to a particular example.

Future work will measure the adaptability of simulation models in other scenarios and using different simulation software packages. From this we will identify the characteristics that make simulation models adaptable. Then, we expect to create simulation software and modeling principles that lead to adaptable simulation models.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation through the Collaboration for Industrial and Systems Engineering: North Carolina A&T State University, University of Maryland,

and Georgia Institute of Technology. The authors thank Mandar Chincholkar and Sean Gahagan for their help with the example.

REFERENCES

1. Shi, Y., and M. Gregory, "International Manufacturing Networks - to develop Global Competitive Capabilities", *Journal of Operations Management*, Vol. 16, No. 2-3, p 195-214, May 1998.

2. NIST, "Modelling and Simulation Environments for Design, Planning and Operation of Globally Distributed Enterprise", proposal document for recently initiated global project, Novenber 1997.

3. Lee, Jay, "Teleservice engineeering in manufacturing: Challenges and Opportunities", *International Journal of Machine Tools & Manufacture*, Vol. 38, No. 9, p 901-910, August 1998.

4. Lilegdon, W. R., "Simulation for Planning and Scheduling - More than What If," 1997 American Production and Inventory Control Society Annual International Conference, Washington DC. December 1997.

5. Odhabi, H.J., R.J. Paul, and R.D. Macredie, " Making Simulation More Accessible In Manufacturing Systems Through A 'Four Phase' Approach", 1998 Winter Simulation Conference, Washington DC. December 1998.

6. Mackulak, G.T., and F.P. Lawrence, " Effective Simulation Model Reuse: A Case Study of AMHS Modeling", 1998 Winter Simulation Conference, Washington DC. December 1998.

7. Balci, O., C. Ulusarac, P. Shah, and E. A. Fox, " A Library of Reusable Model Components for Visual Simulation of the NCSTRL System", 1998 Winter Simulation Conference, Washington DC. December 1998.

8. Jerry Banks and Randall Gibson, "Simulation Evolution," *IIE Solutions*, November, 1998, pages 26-29.

9. Banks, Jerry, ed., *Handbook of Simulation*, John Wiley and Sons, New York, 1998.

10. Law, Averill M., and W. David Kelton, *Simulation Modeling and Analysis*, 2nd edition, McGraw-Hill, New York, 1991.

11. "1999 Simulation Software Survey," *OR/MS Today*, Volume 26, Number 1, pages 42-51, 1999.

12. Catherine M. Harmonosky, "Simulation-based real-time scheduling: review of recent developments," *Proceedings of the 1995 Winter Simulation Conference*, Alexopoulos *et al.*, eds., pp. 220-225, 1995.

13. Herrmann, J.W., C.-Y. Lee, and J. Hinchman, "Global job shop scheduling with a genetic algorithm," *Production and Operations Management*, Volume 4, Number 1, pp. 30-45, 1995.

14. Capers T. Jones, Reusability in Programming: A Survey of the State of the Art, IEEE Transactions on Software Engineering, SE-10, no. 5, September 1984, pp. 488-94.

15. T.C. Jones, Software Return on Investment Preliminary Analysis, Software Productivity Research, Inc., Burlington, MA, 1993.

16. Wayne C. Lim, Managing Software Resue, Prentice Hall PTR, Upper Saddle River, NJ 07458. ISBN 0-13-552373-7.

17. A. Goldberg and K. S. Rubin, Taming Object-Oriented Technology, Computer Language, vol. 7, no. 10, October 1990, pp. 34-35.

18. Wim Codenie, Koen De Hondt, Patrick Steyaert, and Arlette Vercammen, From Custom Applications to Domain-Specific Frameworks, Communications of the ACM, vol. 40, no. 10, Dec., 1997, pp. 70-77.

19. Karl J. Lieberherr Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns, PWS Publishing Company, Boston, 1996.

20. Patricia E. Byrnes, Thomas P. Frazier, and Thomas R. Guliedge, Returns-to-scale in software production: A Comparision of Approaches, Thomas R Guliedge and William P. Hutzler (Eds.), Analytical Methods in Software Engineering Economics, Springer-Verlag, 1993, pp. 75-97.

21. M. Mendelbaum, "Flexibility in Decision-Making: an Exploration and Unification", Ph.D. Thesis, Department of Industrial Engineering, University of Toronto, Toronto, Ont., 1978.

22. J. A. Buzacott, "The Fundamental Principles of Flexibility in Manufacturing Systems", Proc. First International Conference in Flexible Manufacturing Systems, 23-30, North Holland, Amsterdam, 1982.

23. Kenneth N. McKay and John B. Moore, "State of the art scheduling survey," Report R-91-IMM-01, CAM-I, Arlington, Texas, 1991.