SUPPORTING MANUFACTURING WITH SIMULATION: MODEL DESIGN, DEVELOPMENT, AND DEPLOYMENT

Frank Chance

FabTime Inc.
325M Sharon Park Drive #219
Menlo Park, CA 94025
www.FabTime.com

Jennifer Robinson

FabTime Inc.
325M Sharon Park Drive #219
Menlo Park, CA 94025
www.FabTime.com

John Fowler

Arizona State University
Dept. of I.M.S.E., Box 875906
Tempe, Arizona 85287-5906

ABSTRACT

In this paper, we identify and discuss the features we believe are key to the successful use of simulation as a manufacturing support tool. The discussion begins with three sample projects drawn from the authors' industrial and consulting experiences. Using these projects as motivation, we discuss the ideal project lifecycle — model design, development, and deployment. For model design, we emphasize the importance of a clear and consistent specification, articulated in a written document. This specification should identify project customers, goals, and deliverables. We next review a range of model development options, stressing the existence of many non-simulation alternatives. We also discuss methods for model verification and validation. Finally, we consider the difficulties of model deployment, including simulation output analysis, data maintenance, and model integration. We close with several suggestions on how best to present simulation results to a management audience.

1 THREE SAMPLE PROJECTS

To motivate the remainder of this paper, we begin by describing three sample manufacturing projects. These projects are drawn from the authors' experiences.

1.1 Planning Toolset and Staffing Levels

In this project, the client is planning for a new factory, and wishes to compare toolset and staffing levels required to support a variety of different product mix and volume scenarios. The results of the project will be used for three purposes: 1) to determine how many of each type of tool to order; 2) to decide how many operators and maintenance engineers will be needed; and 3) to estimate the relative costs of producing the different products. Because of the high likelihood of product mix changes, the model should allow for quick analysis of multiple scenarios, and should provide sensitivity information. The model will likely be used throughout facility planning and construction.

1.2 Optimizing Cycle Time with Equipment

In this project, the client has already begun building a new single-product factory, and wishes to optimize cycle time by making additional capital equipment purchases. The factory will produce the company's latest product, which is expected to have high demand. A pilot line in an existing facility is currently producing the new product at very low volumes. However, the actual process parameters are in a constant state of flux as engineers make refinements. The client has developed an in-house Excel capacity model. This model is periodically updated with the latest process changes. From this project, the client expects to get a list of additional tools that should be ordered beyond the minimum cost toolset, and an estimate of the expected cycle time once the factory is operational.

1.3 Optimizing Cycle Time with Scheduling

In this project, the client wishes to implement a scheduling system that will balance the competing demands of many different customers for specialized orders. The total production volume is high, and between-product setup times are significant. The client has a sophisticated in-house shop-floor tracking system that is continuously updated with the location of all work in process. The system also has product routing data and equipment status information. The client needs to be able to test out different dispatching strategies, to minimize overall cycle time. Also, the client wishes to be able to track the status of individual orders, so that expected delivery dates can be given to customers based on the latest available information.

2 PROJECT LIFECYCLES

The sample projects outlined above pose questions that can be answered a variety of ways. One such method is physical experimentation — making changes in the real factory and analyzing the results. An example might be a new configuration of a workcell, which is tested out in a small area before being implemented in the whole factory. Physical experimentation is not always possible, however, especially when planning for a new factory. Even in an existing factory, it is often prohibitively expensive to experiment with the real facility. Increasing equipment quantities, for example, can require many months of lead time, not to mention sizable capital investment. Therefore, it is usually necessary to build a model of the real factory, and perform experiments in this virtual environment.

In an ideal world, modeling projects would progress in an orderly fashion through three distinct phases — model design, development, and deployment. In reality, however, these phases often overlap, or are repeated as later phases shed light on errors or omissions in earlier work. Changes in project scope can also cause iterations through these phases. Often, the later work cannot even be clearly defined until the earlier phases are completed. Interaction is usually needed between the client and the analyst to clear up details and discuss new possibilities. However, by following as closely as possible the model design, development, and deployment phases, we believe that it is possible to minimize false starts. In the remaining sections of this paper, we outline some key features that we have found contribute to successful manufacturing simulation projects.

3 MODEL DESIGN

Building and using a factory model can be a daunting task. The sheer size and complexity of most factories makes it difficult to completely view and understand all of the many elements found there. These elements include products, processes, tools, material handling systems, inventory, and operators. We believe that the goal of any modeling project is to develop a model that reflects enough, but not too much, of this complexity. Models with excessive detail take longer to build, debug, and understand, are harder to maintain, and have longer run times than less detailed models. The guiding light in determining how much complexity is necessary should be the questions posed of the model. Clarification and documentation of these questions, then, is of paramount importance, and is the primary goal of the design phase.

3.1 Identifying Project Customers

The first step in the design phase is identifying who will use the results the project is meant to create. Potential customers might be from the shop floor, or from a corporate planning department. In many cases, a project will have several customers, each of whom has his or her own expectations. Prioritizing among the needs of these different customers is the client's internal responsibility. However, the analyst must sometimes consolidate various customer objectives, and identify which of these, if any, conflict.

For example, the customers for the scheduling project outlined above probably include the computer integrated manufacturing (CIM) group and the factory operations manager. The CIM organization might want to completely automate lot selection, while the operations manager might wish for more operator empowerment. Resolving these issues in the product design phase can be much more cost effective than dealing with them after the model is built.

3.2 Identifying Project Goals

As indicated above, a project may have multiple customers, and/or multiple goals. Trade-offs may be necessary. The goals ultimately selected help determine the planning horizon, and should be stated in terms of quantifiable performance measures. The specification of project goals should also include a project timeline with deliverables highlighted. The three goals in our first example are clearly stated in Section 1.1. The goal in the second example is to optimize the cycle time of the new factory by adding equipment. However, more specific information such as how much the cycle time needs to be reduced, or how much money can be spent to reduce it, is also needed. The third example, the shop floor scheduling model, is a potentially huge project that requires the definition of smaller, tangible goals.

3.2.1 Identifying the Planning Horizon

The planning horizon to which the model will be applied is important in determining the level of detail to include. A common classification of planning horizons is strategic, tactical, and operational. Although no formal definition of these horizons exists, strategic models are usually concerned with long-term questions like "should I build a new factory, and if so, what should I produce in it?" The first two examples in this paper are primarily strategic questions. Tactical models address questions like "how many units should I start this quarter?" Operational models are concerned with production over the next day or week. The third example in this paper, the scheduling system, is mainly an operational model, though questions such as "what is the best dispatch rule to use every day?" might be considered tactical.

The way in which a model adds value is highly influenced by the project's planning horizon. For example, at the strategic level the analyst has the opportunity to make recommendations that save (or cost) the company millions of dollars in equipment purchases. In this case, there is a large value to answering these few key questions correctly. At the operational level, the analyst has the opportunity to set in place a system that will help with many day-to-day or hour-to-hour decisions ("what job should I run next on this machine?" or "when should I promise this order will be shipped?"). Here, there is a large cumulative value to making these many small decisions correctly.

3.2.2 Identifying Performance Measures

Performance measures are necessary to quantify project goals. Performance measures should be tracked in the real world, not just in models. The performance measures in the first example will likely be expressed primarily in terms of cost, both toolset cost and product cost. The primary performance measure in the second example is cycle time, though toolset cost is also important. Performance measures in a scheduling system might include percentage of on-time delivery, product cycle time, or average work-in-process (WIP).

3.3 Writing a Project Specification

Ideally, every project should have a document that provides a black-and-white synopsis of project design. This document should identify the customer or customers, the planning horizon, the project goals (stated in terms of quantifiable performance measures), and the expected deliverables. Where possible, the expected deliverables should be expressed in the form of mock results charts. The analyst is then responsible for filling in the charts with actual model output. This helps to ensure that the results provided from the project are in the format that the customer needs.

For example, Figure 1 displays a mock results chart for the first sample project (planning toolset and staffing levels with mix/volume uncertainty). This chart displays two performance measures, toolset cost and annual profit, for four candidate scenarios. Figure 2 displays a mock results chart for the second sample project (cycle time optimization via capital purchases). The third sample project would have a mock results chart similar to Figure 1, except the performance measures would likely be cycle time or on-time delivery percentage.

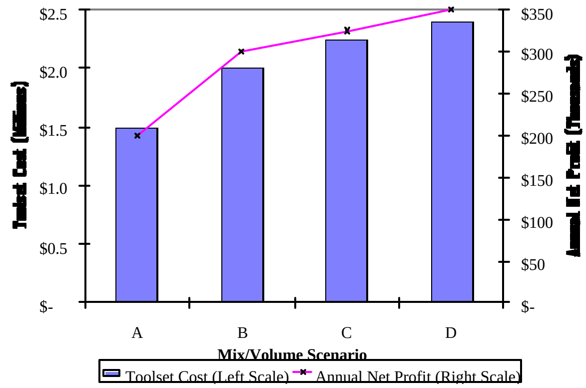

Figure 1: Toolset Cost and Annual Net Profit for Four Mix/Volume Scenarios

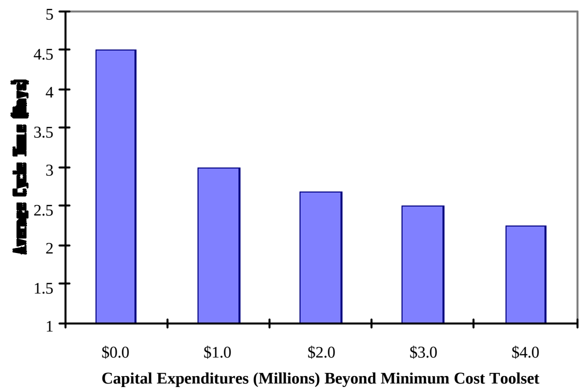

Figure 2: Average Cycle Time vs. Additional Capital Expenditures

In addition to summarizing what the analyst will deliver, the specification should also include client responsibilities. In particular, the specification should document what data the client will provide, and in what basic format (electronically, on paper, in spreadsheets, etc.). In our experience, misunderstandings about the quality of the data available, and who is responsible for improving it, can result in considerable project delays.

The specification should be a living document. As the project scope or deliverables evolve, the specification should be updated to reflect these changes and their impacts. These changes, however, require that both the client and the analyst are in agreement, and the specification often serves as a good negotiation tool. While the specification should not be changed radically in the midst of the project, adding specific details and refinements as they become apparent is beneficial.

4 MODEL DEVELOPMENT

Once the project specification has been written, model development can begin. This requires selecting a methodology and tool, building the model, populating the model with the necessary data, and verifying and validating both the model and the data. Decisions in this phase of the project depend in particular on whether the model is intended for single use or for on-going support. For the latter, the question of who will actually own and maintain the model is important. Also of obvious importance is the type of question being answered by the model. Certain modeling methodologies are more appropriate for certain types of questions. In some cases, what types of tools are available can drive the questions that are asked in model design. In other cases, the client may specify the use of a particular tool or methodology. In general, we feel that the client should identify questions to be answered, while the analyst should be primarily responsible for actual model selection and development.

4.1 Selecting a Methodology and Tool

Many different methodologies and tools are available to support manufacturing projects, including spreadsheets, analytic models, data-driven simulation models, simulation languages, and general-purpose programming languages with simulation libraries. Some projects may require the use of more than one tool to answer different types of questions. Some strengths and weaknesses of these tools for different applications are discussed below.

Spreadsheet models are appealing for projects where a large number of scenarios must be evaluated quickly. They are fast, easy to understand and use, and easy to modify. Managers are generally accustomed to reviewing results presented in spreadsheet-based formats. However, there are limits to the types of questions that can be addressed with spreadsheet models. Spreadsheet models are static, generally grouping time into large buckets. They are also usually deterministic. Though spreadsheets typically include the ability to sample from distributions, this capability is not usually exploited in manufacturing applications. Because of these limitations, spreadsheet models are often not capable of estimating cycle time or WIP.

Analytic models include capacity analysis, queueing, linear programming, and other math programming models. For an example of the successful application of linear programming-based methods to manufacturing production planning, see Leachman et al. (1996). Like spreadsheet models, analytic models are useful for evaluating multiple scenarios, because they are typically very fast. Analytic models can also include detail beyond that of spreadsheets. For example, capacity analysis models can accurately capture complexity such as rework, batch processing, and re-entrant flows. Queueing models can improve upon spreadsheet models by including variability with arrival and service time distributions. Most of the results available in the area of queueing models, however, are only applicable to long-term, steady state behavior. Queueing models can thus be used to estimate long-run average cycle times and WIP, but not short-term behavior (although there is active research in this area, we have not seen any commercial applications of the methodology to date).

The results from analytical models are relatively easy to interpret, because they consist of a single number for each performance measure. However, the simplifying assumptions necessary to get closed form results are not always appropriate, especially in queueing models. Analytic models are best suited to strategic and tactical questions, where the emphasis for dynamic behavior is on relative performance.

Discrete event simulation can capture virtually any level of manufacturing detail, and is potentially very accurate. It captures both dynamic (time-dependent) and stochastic (random) behavior. Simulation is sometimes used in the early stages of a project to help understand how the system works. It has intuitive appeal for managers, especially when animation is used, because they can ‘see’ what is going on in the model. However, there are several disadvantages to using simulation models. They typically take much longer to run than analytic models, and the results from a simulation model can be difficult to interpret. Statistical analysis of the output is necessary, because each simulation run represents a single possible sample path. For these reasons, simulation can be an expensive option.

When simulation is chosen for a project, it is also necessary to decide between data-driven simulation models (simulators), and user-developed models written in a simulation or general-purpose programming language. In our opinion, languages are more appropriate for building small, detailed models, while simulators tend to be more applicable for modeling large-scale manufacturing systems. The decision also depends on the modeler’s proficiency with different tools, and on whether or not the model will be reused.

Overall, tool choice for a project should be most heavily influenced by the needs of the client, particularly if the model is to be used on an on-going basis after the project is complete. Attention must be paid to what the client will be able to learn, use, and modify as needed. The planning horizon and the project goals will also influence which tool is appropriate. Clearly, if accurate cycle time estimation is critical, a spreadsheet model is probably not the best choice. If many different scenarios need to be evaluated quickly, with comparisons based on toolset capacity and cost rather than on cycle time, simulation may not be the best choice. We think that it is important to remember that simulation is not the only tool available. In many cases, simulation may not be required at all. The goal of the project is to answer the questions posed by the client, using the simplest model.

For the first project in this paper, we would recommend a spreadsheet or capacity analysis model. The second project is better suited to a hybrid capacity analysis and simulation model. The third probably requires a simulation for evaluating potential dispatch rules. Analytic models might also be embedded in short-term dispatch support systems.

4.2 Developing the Model and Collecting Information

Once the tool and methodology have been selected, the basic structure of the model should be constructed. This requires gathering information about the real system, including things like the total number of products, the number of tools, the general sequence of steps followed by operators at the different machines, and the structure of the material handling system. The analyst should rely on the project specification to resolve disputes about what the model should be able to do and what it will not be expected to do. Tools that grow haphazardly over time, as different functionalities are included in response to input from various sources, are extremely difficult to use. At this stage, it is important to keep in mind the overall goals of the model. The actual detailed steps followed in model development depend on the tool being used.

4.3 Populating Model Data

Perhaps the biggest roadblock to building models for real manufacturing applications lies in the difficulty of obtaining the proper data. In our experience, there are three classes of data: 1) the data you want; 2) the data you need; and 3) the data you get. These sets are typically decreasing in size, but are not necessarily overlapping. That is, data may be provided that is not needed for the model, even as data critical to building the model is unavailable. One suggestion is to find out what types of data will be available before spending much effort designing and developing the model. In many cases, the accuracy level of the data does not justify a complex model. If some data is simply not available, it may be possible to perform sensitivity analysis to see how important the missing data is. Only if the data will likely have a significant impact on the model are new data collection efforts necessary. Carrying this approach to its extreme, it is sometimes advantageous to build a model with a bare minimum of actual data. At that point, sensitivity analysis can be used to determine the priority areas for data collection.

Of our three sample projects, only the third is likely to have very detailed, accurate data. Of course, even with a detailed shop floor control system, some information may not be entered into the system at all, or may be entered inaccurately. In some manufacturing facilities, detailed information is only kept as it relates to the bottleneck pieces of equipment. In such cases, it may make more sense to only build a detailed model of the top few bottlenecks. For the other two projects, it is likely that any existing data will be stored in spreadsheet form. Often, data for the same factory will be stored in a variety of spreadsheets, and these spreadsheets will be maintained by several different employees.

4.4 Verifying and Validating the Model

We distinguish between model verification and model validation as follows: verification is the process of ensuring that the model produces a correct output given a specified input, while validation is the process of ensuring the model accurately represents reality to the necessary degree (100% accurate representation is usually not required for a successful project).

For verification purposes, we recommend beta-testing the model early and often. These tests should attempt to produce solid, repeatable results with the smallest possible set of input data. One way to assist this process over time is to build up a library of verification tests, the expected results of which are known in advance. Whenever possible, such verification suites should be automated. This reduces any temptation to cut corners. Each model should also be stress-tested with a subset of actual data from the facility being modeled (if possible). The analyst should make changes to the input data and examine the results to ensure that outputs move in the expected direction, and that no counter-intuitive situations arise. Counter-intuitive results can be significant, however, since they sometimes provide important new insights into system behavior.

4.5 Verifying and Validating the Data

As with model beta-testing, we recommend starting data verification and validation efforts as early as possible in the modeling process. An example of data verification might involve having end-users check data forms or data summary charts to see that data has been entered correctly. An example of validation would be reviewing outputs with end-users to ensure that results look reasonable. Often, it will take numerous passes before bugs in the data are completely worked out. For existing factories, another method of validation is a variant on the Turing test (see Schruben 1980). This involves taking model output data and transforming it into the same format as typical manufacturing reports. If manufacturing personnel cannot distinguish between the model output and actual factory output, then the model is probably a reasonable representation of reality. Tool utilization, cycle time, and production volumes are possible candidates for this method, as they are likely to be included on existing production reports. In some cases, there may be other models that can be used for validation. For example, in our second project, a new capacity and simulation model could be validated against the existing spreadsheet-based capacity model.

5 PROJECT DEPLOYMENT

Frequently, model development and debugging consume the lion's share of the project schedule. This is in spite of the fact that the model itself is not the project's end goal. The goal is to use the model to answer the questions outlined in the project specification. For most simulation projects, this requires analyzing the simulation output, presenting the results to management, and setting up the model for on-going data collection and analysis (if needed). These topics are outlined below, followed by a brief discussion of common simulation project pitfalls.

5.1 Analyzing Steady-State Simulation Output

Often, simulation projects do not require steady-state analysis. When required, however, steady-state analysis presents a variety of difficulties not found in finite-horizon analysis. Simulation texts usually treat this topic in depth (see, for example, Law and Kelton 1991). Nelson (1992) also provides a helpful overview of steady-state analysis. Two features often found in manufacturing simulations cause special difficulties — highly correlated output data and initialization bias.

Highly correlated data makes it difficult to produce within-run confidence intervals using traditional statistical methods. If the correlation is not taken into account, the width of confidence intervals is usually biased on the low side. This bias can cause modelers to predict significant differences between alternatives where none exist. It may be possible to batch the output data in such a way as to reduce the serial correlation to a negligible level. Otherwise, the best way to avoid the problem of correlated data is to make independent replications and use cross-replication results to generate confidence intervals. Using multiple replications, however, can aggravate initialization bias problems.

Initialization bias is present in any simulation where the initial simulation state is not drawn randomly from its steady-state distribution. Since this steady-state distribution is never known exactly (otherwise, the problem would be trivial), and may not exist at all, initialization bias is present to some extent in nearly all steady-state manufacturing simulations. Since this bias usually dies away over time, one popular method is to run the simulation for a very long time and then truncate the first portion of the output. Another option is to initialize the factory with a certain amount of work-in-process inventory. With the former method, the difficulty lies in deciding how much to throw away; with the latter method, it lies in deciding initial WIP levels.

Neither method is entirely satisfactory, but two qualitative observations are in order. First, the best way to determine if initialization bias is a problem is to look at time series output. For example, Figure 3 displays time series output for cycle time in a manufacturing simulation. From this chart, it appears there is negligible initialization bias. Figure 4, displays cycle time output that appears to contain significant initialization bias.

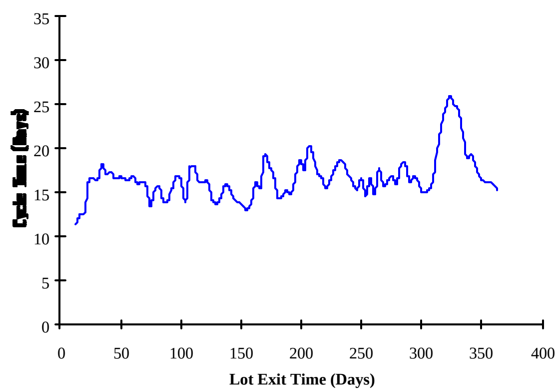

Figure 3: Cycle Time vs. Lot Exit Time for Run with Negligible Initialization Bias

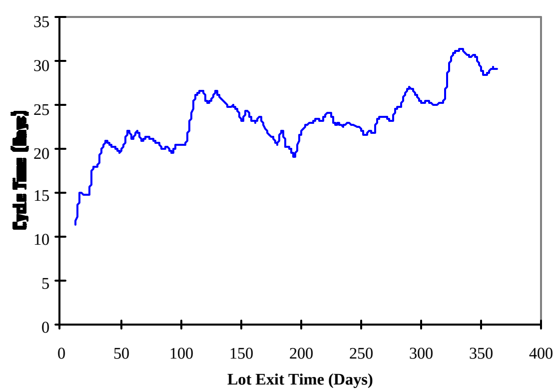

Figure 4: Cycle Time vs. Lot Exit Time for Run with Significant Initialization Bias

Our second qualitative observation is that the effect of initialization bias in manufacturing simulations generally increases as utilization rises. Thus, a truncation point that successfully eliminates most initialization bias for a highly loaded system will generally work quite well for more lightly loaded systems. In fact, the two series shown above are from simulations of the same model. Figure 3 displays output from a 65% utilization run, while Figure 4 displays output from a 95% utilization run. Not only does the second run have a higher cycle time, as we would expect, it also appears to have a longer transient effect.

To set a truncation point, visually inspect output charts of data averaged across multiple replications. This averaging will smooth out random fluctuations and will make the trend easier to pick out. For example Figure 5 displays the cycle time output series for 95% utilization averaged across ten replications. From this chart, it appears that a truncation point of six months or more is required. At this point, we would recommend making multiple replications of somewhat longer runs to confirm that six months is in fact a reasonable truncation point.

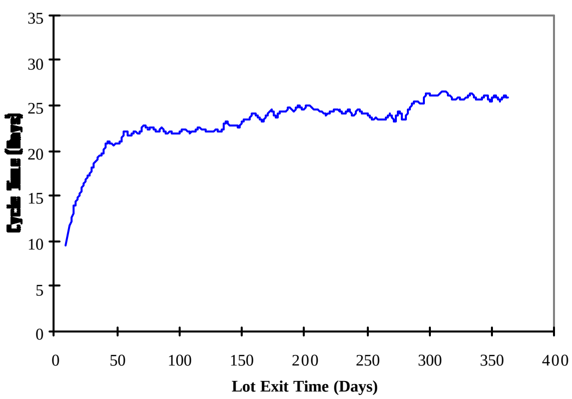

Figure 5: Cycle Time vs. Lot Exit Time Averaged Across 10 Independent Replications

When it is not possible to visually examine all relevant output, one can employ a statistical test, such as the one described by Schruben, Singh, and Tierney (1983).

5.2 Presenting Simulation Results to Management

When presenting results to management, there are several things to keep in mind that can increase the likelihood of model acceptance. First, display results graphically whenever possible. This makes it easier to understand trade-offs in a short amount of time. The graphs should follow the formats identified in the project specification. Second, present only a few vital pieces of output. Do not waste managers' time by forcing them to wade through a myriad of details. In particular, make sure that the results presented answer the questions asked in the original project specification. Finally, whenever possible present results in terms of dollars. Managers have a ready grasp of the bottom line. And, ultimately, project success will be judged in terms of tangible financial results.

5.3 Maintaining an On-Going Model

Considerably more work is required to build a model that will be used on an on-going basis. Such a model must be robust enough to be used by different people, and flexible enough to be changed over time. Systems that will be used on a daily basis will probably have to be integrated directly into existing systems. For example, a scheduling system, to be effective, must receive at least periodic updates from the shop floor control system. A strategic planning model, such as the one identified in our first example, is more likely to exist as a stand-alone model. A model like the one in the second example, with a single process flow from an existing product, could be integrated with the Excel capacity planning model. This would have the advantage of being able to automatically update the simulation model when changes were made to the Excel model. If the model is really only going to be used once, such integration is probably not worth the effort. However, models intended for a single use often end up being used on an on-going basis. This can lead to much more work than integrating the models in the first place, and can be the source of many data maintenance nightmares.

5.4 Common Simulation Project Pitfalls

Simulation projects can fail for a variety of reasons. Perhaps the most common problem is lack of a good project design. Models are built without regard to the questions that they will answer, and then they are not able to adequately provide the answers that are needed. Models built for single-use often end up being maintained and used over time. If they have not been designed to be used by someone other than the model developer, inaccurate results can be generated. Lack of a clear design can also result in insufficient buy-in and participation from the client.

Another common problem is data. Sometimes the necessary data does not exist, or is prohibitively expensive to obtain. Other times inaccurate data is used through oversight, or lack of clear lines of responsibility for collecting the data. A prevalent problem with simulation models is that they are too detailed. As a result, they are not maintained, because data collection is such an effort, or they are not run at all because the runs take such a long time.

A more subtle reason why simulation projects sometimes fail is that the performance measures optimized in the model are different from the performance measures rewarded in real life. For example, few simulation models will be able to motivate workers to reduce cycle time when pay standards are based upon throughput.

In general, we believe that more of these problems stem from project management / environmental issues than from lack of simulation expertise on the part of the analyst.

6 SUMMARY

Despite the cautionary tone we have taken in this paper, we believe it is possible to successfully use simulation in a manufacturing environment. Based on our experience, however, doing so requires more than just expertise in simulation. First and foremost, it requires that the analyst work with the client to prepare a written project specification. This document should clearly spell out project customers, goals, and deliverables. Once goals are established (along with suitable performance measures), the analyst should consider the entire range of available analysis tools. Depending on the project goals, a non-simulation methodology may be preferable. Even within the class of simulation tools, there is usually a spectrum of choices ranging from data-driven simulators to full-blown simulation languages. No matter the choice of tool, data-related problems should never be ignored or underestimated, as they often cause significant delays in model development. Model completion, however, does not guarantee project success. That comes only when the model is used to answer the questions it was built to satisfy. Then, and only then, will simulation be viewed by management as an effective decision-support tool.

REFERENCES

Law, A. M., and W.D. Kelton. 1991. *Simulation Modeling and Analysis*. McGraw-Hill.

Leachman, R., R. F. Benson, C. Liu, and D. J. Rarr. 1996. IMPReSS: An Automated Production Planning and Delivery Quotation System at Harris Corporation - Semiconductor Sector. *Interfaces* 26#1 (Jan/Feb): 6-37.

Nelson, B. L. 1992. Statistical Analysis of Simulation Results. In *Handbook of Industrial Engineering*, 2nd Edition: 2567-2593.

Schruben, L. W. 1980. Establishing the Credibility of Simulation. *Simulation* 34: 101-105.

Schruben, L. W., H. Singh, and L. Tierney. 1983. Optimal Tests for Initialization Bias in Simulation Output. *Operations Research* 31: 1167-1178.

AUTHOR BIOGRAPHIES

FRANK CHANCE is President and co-founder of FabTime. He has been developing software since 1983, and performing factory productivity improvement consulting for the semiconductor industry since 1993. Immediately prior to founding FabTime, Dr. Chance was co-founder and product manager for C2MS Productivity Solutions, a provider of productivity improvement software to state government licensing agencies. Before that, he founded Chance Industrial Solutions (CIS), a company that provided software and consulting services for the semiconductor industry. CIS developed and commercialized Factory Explorer®, an integrated factory-wide capacity, cost, and simulation analysis tool. From 1994 to 1995 Dr. Chance was a visiting assistant professor at the University of California, Berkeley, Department of Industrial Engineering and Operations Research. From 1993 to 1995 Dr. Chance was a consultant to the SEMATECH, where he developed Delphi, the primary fab capacity analysis and simulation tool used by the Measurement and Improvement of Manufacturing Capacity project. Dr. Chance has dual AB degrees in Mathematics and History from Washington University. He also has M.S. and Ph.D. degrees in Operations Research and Industrial Engineering from Cornell University.

JENNIFER ROBINSON is Chief Operating Officer and co-founder of FabTime. She has been offering productivity improvement services to the semiconductor industry since 1992. Immediately prior to founding FabTime, Dr. Robinson was co-founder and senior analyst for C2MS Productivity Solutions, a provider of productivity improvement software to state government licensing agencies. Before that, she consulted in factory productivity analysis for the semiconductor industry for several years. Her clients included Digital Equipment Corporation (now Intel Corporation), Siemens AG, Seagate Technologies, IBM, Wright Williams & Kelly (WWK), and SEMATECH. For SEMATECH, she developed and delivered (as part of an international team) a series of two-day courses on measuring and improving wafer fab capacity. Dr. Robinson has a B.S.E. from Duke University, an M.S. in Operations Research from the University of Texas at Austin, and a Ph.D. in Industrial Engineering from the University of Massachusetts at Amherst.

JOHN W. FOWLER is an Assistant Professor in the Industrial and Management Systems Engineering Department at Arizona State University. Prior to his current position, he was a Senior Member of Technical Staff in the Modeling, CAD, and Statistical Methods

Division of SEMATECH and an Adjunct Assistant Professor in the Graduate Program in Operations of the Mechanical Engineering Department at the University of Texas at Austin. He received his Ph.D. in Industrial Engineering from Texas A&M University and spent the last 1.5 years of his doctoral studies as an intern at Advanced Micro Devices. His research interests include modeling, analysis, and control of manufacturing (especially semiconductor) systems. He is a member of IIE, INFORMS, and SCS.