Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

USE OF DISCRETE EVENT SIMULATION TO ANALYZE DISPATCH POLICIES OF AN EQUIPMENT GROUP IN SEMICONDUCTOR FAB

Raja Sunkara

Manufacturing Systems Engineer
Corporate Information Services
National Semiconductor Corp.
2900 Semiconductor Drive
Santa Clara, CA 95051, U.S.A.

Ramesh Rao

Manager, Manufacturing Decision Tools
Manufacturing Engineering
National Semiconductor Corp.
1111 West Bardin Road
Arlington, TX 76017, U.S.A.

ABSTRACT

This article describes a methodology to model complex operation and process driven practices using a discrete event simulator. This level of detail in the model is critical for the analyses and design of complex operation and process driven dispatch policies in a semiconductor fab. The modeling of these practices is typically not a part of the general set of rules and methods provided by commercially available simulation software. The methodology provides key information that simplifies the development of suitable dispatch policies subject to factory dynamics. The modeling philosophy plays a key role in the success of simulation as a culture. As an example, we present the modeling of complex floor practices to analyze the impact of setup changes subject to process restrictions.

1 INTRODUCTION

Discrete event simulation modeling has become a widely recognized management tool by many manufacturing firms. Several companies have endorsed policies requiring some form of simulation evaluation before approving and committing new investments on production resources (Lung 1998). Many companies use simulation to address classical problems such as production bottlenecks, shop floor layout, material transportation, capacity balancing and cycle time planning but very few use simulation for the design, modification, and improvement of work processes (Melão and Pidd 2003). Despite the general consensus that simulation is a powerful manufacturing system analysis tool, simulation models are usually developed as a one-time use analytical model except in cases where the simulation model is used for simulation based control (Smith et al. 1994) or simulation based scheduling (Harmonoski 1995). Typically, simulation is not used on a regular basis due to the time and resource commitments required for the construction and maintenance of an accurate simulation model. While most organizations use traditional "IE group" approaches to data-acquisition, maintenance, model building/analysis, approaches to automate model generation and provide interactive analysis have also been developed (Son, Jones and Wysk 2000; Kulvatunyou and Wysk 2001). Irrespective of the modeling approach, successful use of simulation as a manufacturing system analysis tool on a daily basis requires complete support from upper management and active participation by the entire organization.

National Semiconductor's fab in Arlington Texas (NSTE) has been using simulation as a manufacturing system analysis and planning tool for over five years. NSTE is a high volume wafer-manufacturing site producing a wide mix of products on many different types of technologies (CMOS, BICMOS, BIPOLAR, etc.). Management requires that all capital plans to purchase tools be justified by simulation analysis prior to approval. Irrespective of the loadings and product mix there are very stringent cycle-time requirements. The model is owned by Manufacturing and trusted by Management. Manufacturing uses the model on a daily basis for analysis and planning. Simulation output is used to drive Real Time Dispatching (RTD) rules and to set daily plant production goals including detailed goals down to the equipment level (Appleton-Day and Liming 1997). This level of commitment is possible due to complete support from management. The entire organization is held responsible for the accuracy of the simulation data and is fully engaged to make simulation the culture.

The simulation infrastructure used at NSTE is geared towards a mature fab with an equipment base spanning several generations of tools with little or no automation. As a result, it is not possible to set up automated data collection directly from the tools and a process for keeping the data updated needs to be established. It is also extremely important that the data and the model be validated regularly. The simulation framework and modeling process play a crucial role and are briefly presented in subsequent sections.

Due to the different technologies and constantly shifting product mix, scheduling and dispatch rules need to be dynamic. The complex interactions between product mix, tool dedication, and process restrictions mandate a detailed simulation model to evaluate dispatch and scheduling policies. In addition to traditional scheduling and dispatching policies, we also use the simulator to determine policies guiding setup changes, tool configurations for clustered tools (ASM-FSI Photo Clusters), operator/operator certification requirements, etc.

As with any simulation approach it is imperative that the model reflects reality to the extent that the results and conclusions are meaningful. Some of the detailed modeling required to meet NSTE's needs is not supported by the standard set of rules and methods provided by commercially available simulation software. Hence, it is important that the simulation software support some form of customization. It is equally important that a methodology to add user-defined customizations be developed. This paper presents the approach used by NSTE in this regard.

The remainder of the paper is organized as follows: Section 2 provides more details on the issues addressed. Section 3 describes the simulation infrastructure and modeling process used at NSTE. Section 4 discusses our choice of simulation software and its advantages and limitations. Section 5 discusses user-defined enhancements, why we need them, our implementation approach, and functionality captured. Section 6 outlines the benefits derived. Figures and tables are presented in Section 7. We conclude the paper with a brief discussion on future direction.

2 PROBLEM STATEMENT

Manufacturing at NSTE is heavily dependent on RTD. Besides implementing fab wide dispatch rules to control average cycle-time, RTD is also used to maintain 98th percentile cycle-time metrics, maintain line balance, ensure timely processing of hot lots, maximum utilization of constraint tools, increase utilization of batch tools, etc. Due to the complex nature of a semiconductor fab and the key role that RTD plays at NSTE, management requires that the impact and effectiveness of complex RTD rules be assessed, using simulation, prior to deployment on the floor. Simulation output is also used to drive RTD decisions. As a result, existing dispatch rules/policies used on the floor also need to be represented in the simulation model.

In order to decrease the turn-around time on the development and testing of new dispatch policies, we have developed a framework that simplifies the creation of dispatch policies for the simulation engine. As an example of a complex dispatch rule, consider the following requirements for a dispatch rule in implant:

- Total of 8 implanters (5 belonging to family A, and 3 belonging to family B)
- Of the 4 possible setups (S1, S2, S3, S4) for Family A, S1 is allowed on two tools and S2 is allowed on the other 3 tools. S3 and S4 are allowed on all tools.
- A tool that is setup for S2 has to be setup for S3 before it can be setup for S4
- Not all products can be processed on Family B.
- Process material on Family B only if the workload on Family A would violate the maximum allowed queue time
- Tools in Family A achieve maximum throughput if we can stage multiple lots in front of the tool
- Minimize setup changes
- Maximum time a lot can wait for a tool to switch to the required setup is 14 hours
- If no material requiring a tools current setup is expected to arrive in the next 4 hours then change the setup for the tool to process current WIP
- For setup S3 there are two types of wafers (R and NR). The tool needs to run dummy wafers when switching from the R type wafers to the NR type wafers. Minimize the use of these dummy wafers
- The rule cannot be hardwired for a certain product mix but needs to self balance as the mix changes.

In addition to simplifying the development of dispatch policies, the simulation model needs to be kept up to date with the latest process, product mix, and equipment changes. The modeling process and infrastructure at NSTE has made this a reality in an environment where automated data acquisition is not possible.

3 NSTE SIMULATION INFRASTRUCTURE

Fab operations at NSTE are divided into 7 functional areas. Engineers in each area are responsible for maintaining the simulation data for their area. Each engineer is responsible for the data related to the processes and equipment that they own. All manually collected data is stored in Excel workbooks (referred to as the data books) with a well-defined format and structure. The data books for each area are designed to meet the specific needs of that area. Each area has separate Process, Equipment, Setup, and process overrides data books. Part of the sign-off loop for engineering/process change control includes updating effected simulation data.

Route definitions, Equipment States, Current WIP, Operator certifications are automatically downloaded from the Manufacturing Execution System(MES). Lot start information is provided by the production control department in a data book with a pre-defined format. It is important to note that the format for the data books are independent of the simulation engine and are designed in a form that the engineers can understand. The engineers do not have any training or understanding of the simulator input formats.

At the core of the simulation process is a home grown model building utility referred to as the model builder. All the business rules and the rules governing the translation of the data from the data books to the simulation inputs is stored and designed into the model builder. The model builder automatically reads the data from all input data books (roughly 32 Excel books each averaging about 20 to 25 worksheets). The model builder first checks the data for completeness, format, and limits. Next, the data from the data books is linked to the route definitions from the MES (WorkStream) and the entire set of input data required by the simulation engine is generated.

The simulation outputs are uploaded into global databases and several canned reports are generated. Each time the model builder is executed a detailed error report is generated for each worksheet in each data book. The model builder runs every hour and the error reports are made available over the intranet. The model builder also generates all data required for static analysis.

There is one full time person whose main role is simulation analysis. This person is not responsible for data collection or for creation of the dispatch rules in the simulation engine. The main role of the simulation analyst is to run what-if scenarios, set daily goals, run simulations for capital planning and ramp scenarios, coordinate the efforts of the data owners and educate the data owners on simulation related issues. The simulation analyst also performs model validation and correlates simulated data with historical data. Another source of model validation are automated reports that report any instance where a tool on the floor processed material that was not allowed in the simulator.

The managers and engineers for each functional area are held accountable for any discrepancies between the goals set by the simulator and actual performance on the floor. The simulation analyst validates the models used to generate data for the RTD engine.

The key difference between the traditional "IE Group" approach and the approach taken at NSTE is in the data collection, data/model maintenance, and reporting process.

In the traditional approach the "IE Group" is responsible for data collection, time studies, data/model maintenance, and simulation analysis and reporting. In the NSTE approach data collection and maintenance is the responsibility of the customers, i.e., the engineers, supervisors, and managers who receive the benefits of simulation analysis. This increases ownership and confidence in the analysis. The simulation analyst plays the role of a consultant, helping the customer understand what data to collect and how to go about the data collection depending on the tool type and the goals of the study.

In the NSTE approach simulation output is loaded into global databases and the customer performs most reporting and analysis. Our reporting tool of choice is Business Objects, which is used by all managers and engineers to create their own reports for simulation and other data sources. For more complicated analysis the customers work with the simulation analyst to develop the reports and analysis needed.

All model building activity is performed through the model builder utility. A person with an operations research and software background maintains the model builder utility. This individual is also responsible for the development/implementation of the dispatch rules, software enhancements to the model builder, development of extensions, and overall responsibility of the simulation and dispatch process. All web and database related applications to support the simulation process, reporting, as well as, support to manufacturing on several productivity improvement projects is provided by a third member of the team whose expertise lies in databases, web applications, and software development. The team at NSTE works closely with the corporate information services group on projects such as the development of the framework for modeling dispatch and scheduling policies as outlined in this presentation.

The simulation infrastructure is illustrated in Figure. 1. The automated exception reports are posted on a daily basis and address issues such as the impact of route definition changes on model data, process recipes missing from the simulation data, undefined dispatch stations, and a detailed error report. The input data error report is published on an hourly basis. In addition to the AutoschedAP simulator, we also make extensive use of the Brooks-PRI APF reporter/RTD products. The APF reporter is used for extracts from the MES, analysis of the simulation data, and also serves as the static model for quick analysis. The validation tools include a correlation application (home grown), the APF reporter, and Business Objects. Reporting tools include a homegrown model reporting utility for canned reports, Business Objects for (customer developed) customized reports, and the APF reporter. We also use the AutoschedAP reporting utility for Gantt charts and debugging.

4 SIMULATION SOFTWARE

The simulation software used at National Semiconductor Corporation is the Brooks-PRI AutoSchedAP simulation package (Brooks Automation Inc. 2001a). There were sev-

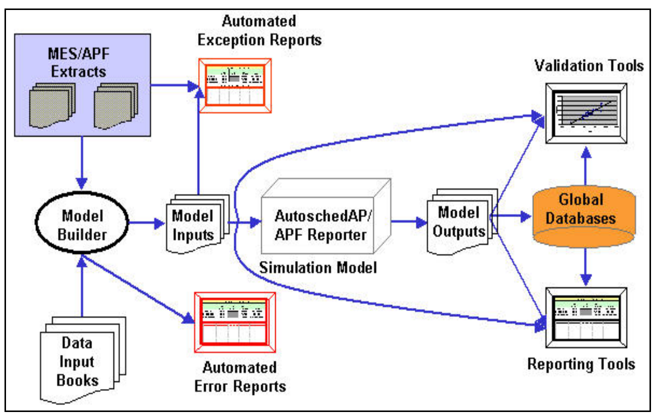

Figure 1: Modeling Architecture

eral reasons why AutoSchedAP was selected as the simulation engine. AutoSchedAP reads all the input data from tab delimited text files, which works well for data maintenance as opposed to a proprietary structured database. The software provides several standard rules and constructs that are geared towards the semiconductor industry. AutoSchedAP provides a flexible framework for user customizations (Brooks Automation Inc. 2001b) that allows users model situations that are specific to individual factories. Customizations are also very well supported by Brooks-PRI. The standard features supporting operator modeling however leaves a lot to be desired. Additional features supporting the development of dispatch rules would be a welcome enhancement.

5 NATIONAL FRAMEWORK DETAILS

The framework for customizations presented in this section was developed by corporate information services in order to decrease the turn-around time on the development and testing of new dispatch policies. The framework provides flexibility in modeling floor rules and provides better analysis and information for decision-making. AutoschedAP provides a framework for the development of additional functionality; the National framework is built on top of the AutoschedAP framework to customize dynamic simulation data to meet the rule development and simulation reporting needs.

5.1 National Framework Design

AutoSchedAP is built using C++ class libraries. The AutoSchedAP framework forms the basis for most of the standard functionality in the simulation. AutoSchedAP provides many simple ways to customize features of the simulation engine, which does not require C++ development. Examples include optional fields in the model inputs and action lists, which allow the definition of a sequence of actions to be taken at different points in the simulation to attain a particular behavior. But some of the sophisticated enhancements like scheduling algorithms and dispatch policies will require additional C++ class library be developed.

The C++ class library developed in the case presented in this paper is referred to as National framework, see figure 2. The National framework simplifies tracking information required to implement the dispatch rules. This takes into account information related to current factory dynamics. The AutoSchedAP framework provides the mechanisms to subscribe to various events (e.g., state changes, lot selection etc.); this feature was extensively used to update the factory data subject to the factory events. The National framework consists of Factory, Station Group, Station Family, Lot, Route and Setup etc. classes which capture different data elements which are updated subject to certain

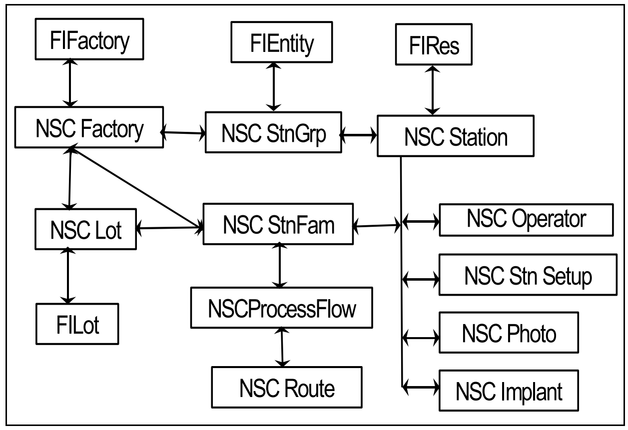

Figure 2: Sample National Framework Classes

events in the simulation. For example, when a lot is released by a station the statistics of the current setup are decremented and the statistics for the next setup are incremented.

5.2 National Framework Functionality

One of the main features of the National framework is to enable dispatching decisions based on the factory status. In this section a subset of the National framework's functionality is described using the implant setup rule, outlined in Section 2, as an example.

The framework provides a mechanism to track current WIP and classify it into several buckets. Some of the dispatching decisions are made looking at the WIP profile of current lots waiting to be processed by the equipment group along with the future lot arrivals. The future arrivals are broken down into hourly buckets for a predefined window of time (usually next 2 or 3 days) to provide finer granularity. This data is available to specific stations of interest as opposed to all stations in the factory to enable faster execution times. The WIP is identified by number of wafers and is further classified based on setup requirements, processing requirements, and process restrictions. WIP belonging to a particular setup is further classified by the wafer characteristics (e.g., R vs. NR wafers in implant). Most of the WIP tracking information is done at factory level for all equipment of interest.

The National framework keeps track of the average wait time for the lots currently in the queue for the equipment under consideration. This helps in enforcing the operational policy of lots be routed to certain preferred group of equipment unless the current queue time at that group is greater than the predefined levels. Individual lot wait times at the current equipment are tracked to take care of process restriction (e.g., a lot can not wait more than 14 hours before next process). The dispatch rule looks at the wait time of each lot and makes an exception if a setup change is required to enforce process restrictions.

AutoschedAP framework publications for an equipment state change event (e.g., process state to setup state) is used to keep note of elapsed time since the last time equipment changed its setup. This metric is taken into account in enforcing another operational policy that equipment cannot run for more than 30 hours on some specified setup.

When a tool finishes processing its current load the next lot selected is based on its current setup. The selection of next lot may trigger a new setup requirement. In order to minimize setups, the dispatch rule selects a lot to work on from the WIP and the framework acts as a mini MES to support dispatching decisions. The potential decisions may range anywhere from leaving the tool idle to changing the tool's setup. In most cases the current setup is changed subject to the WIP profile. In some cases the setup is not changed even though there are no lots waiting at the tool with the current setup requirements. Setup change decisions also take into account number of available tools, the state of each tool (e.g., Down, PM, Qual) and number of available tools with a particular setup. If a tool is in the down state, an estimated remaining down time is taken into account before making the setup decision.

Most of the National framework features described are applicable in general to many of the dispatching situations. The National framework enabled NSTE to rapidly explore other dispatch rules before deploying in the factory. In future the framework may be implemented at other National Semiconductor factories to deploy dispatching rules without redoing the development.

6 BENEFITS DERIVED

The key benefit of the approach described here is the rapid development and testing of dispatch rules using the simulation and the ability to accurately model work processes followed on the floor (driven by many factors including equipment location, process restrictions, availability of information to make real time decisions on the floor, and basic human behavior). Thus, eliminating the need for theoretical factors to compensate for the same.

Increasing the model's ability to represent reality makes it easier to analyze the simulation output since the analyst needs to spend less time approximating the relationship between reality and simulated data. It allows the users to compare simulation results with the real life situation on the floor directly. The flexibility of the framework combined with the NSTE simulation infrastructure enables the simulation analyst to focus on simulation analysis and on exploring what-if scenarios using the model (versus spending time on data collection and data mapping from reality to the simulator and visa versa).

The area managers and supervisors are able to use simulations to make key policy decisions on the floor. In the case of the implant area operations presented here, the range of setup times reported in the simulation model had dropped from a range of 20-25% to 5-9% reflecting the floor operations more accurately. The information and results from this process were extremely valuable in making modifications to simplify the guidelines for managing implant area operations.

7 FUTURE DIRECTION

One of the areas that we are currently working on is to develop generalized tables to input data required to model some of the most commonly used constructs for constraint and bottleneck management, line balancing, and improving batch efficiency. Our estimates suggest that developing a generalized set of tables and implementing the code to interpret these tables as part of the framework will allow us to model about 90% of the requests we receive for RTD without having to modify any code or model structure.

ACKNOWLEDGMENTS

Without the strong management support and commitment from all engineers and managers involved, simulation as a powerful management tool would not be possible. Special mention needs to be made for the contributions made by Aparajit Saigal (database, web based applications, and scripting guru), and, by Sanjay Rajguru (fab Manager) who has been the champion for the Simulation/RTD effort from the beginning.

REFERENCES

Appleton-Day, K., Liming Shao. 1997. Real Time Dispatch Gets Real Time Results, Semiconductor Manufacturing Conference Proceedings, 1997 IEEE International Symposium on , 6-8 Oct. 1997.

Brooks Automation, Inc. AutoSimulations Division, 2001a. AutoschedAP User's Guide v. 7.0.

Brooks Automation, Inc. AutoSimulations Division, 2001b. AutoschedAP Customization Guide v. 7.0.

Harmonoski, C. M. 1995. Simulation-based real time scheduling: Review of recent developments. Proceedings of the 1995 Winter Simulation Conference, 220-225.

Kulvatunyou, B., R.A. Wysk. 2001. Computer-Aided Manufacturing Simulation (CAMS) Generation for Interactive Analysis – Concepts, Techniques, and Issues. In: B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer (eds.). Proceedings of the 2001 Winter Simulation Conference, 968-976.

Lung, A. W. M. 1998. Benefits of a Systematic Approach for Problem Analysis on Manufacturing Simulation Modeling. International Conference on SIMULATION No. 457: 271-276.

Melão, N., and M. Pidd. 2003. Use of business process simulation: A survey of practitioners. Journal of the operational Research Society Vol. 54, No. 1: 2-10.

Smith, J. S., R.A. Wysk, D.T. Sturrok , S. E. Ramaswamy, G. D. Smith, and S. B. Joshi. 1994. Discrete Event Simulation for Shop Floor Control. Proceedings of the 1994 Winter Simulation Conference, 962-969.

Son, Y. J., A.T. Jones, R.A. Wysk. 2000. Automated generation of simulation models from neural libraries: An Example. Proceedings of the 2000 Winter Simulation Conference. 1558-1567.

AUTHOR BIOGRAPHIES

RAJA SUNKARA is a Manufacturing Systems Engineer in Corporate Information Services Division of National Semiconductor Corp. headquarters at Santa Clara, California. He has been working in the semiconductor industry for the past 10 years in the areas of Productivity Improvement, Enterprise Resource Planning, Business Process Automation, and Information Services. He received MS in Industrial Engineering, MBA and BS in Mechanical Engineering. He can be contacted by e-mail at <Raja.Sunkara@nsc.com>.

RAMESH RAO is the Manager of the Manufacturing Decision Tools group and has been with National Semiconductor Corp. at Arlington Texas (NSTE) since 1996. Prior to that he was at the University of Virginia where he received his Ms and PhD Degrees. His interests include simulation, RTD, algorithms, factory analysis tools, e-Manufacturing, and factory automation architectures. You can reach him by e-mail at <Ramesh.Rao@nsc.com>.