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

CONCEPTUALIZATION, DESIGN AND IMPLEMENTATION OF A STATIC CAPACITY MODEL

Orkun Ozturk
Melissa Boom Coburn
Steve Kitterman

Seagate Technology
One Disc Drive
Bloomington, MN 55435, U.S.A.

ABSTRACT

This paper describes the methodology used for development of a static capacity model. It is a well-known fact that no matter how sophisticated the dynamic models are, there is always a need for the simple spreadsheet model. The spreadsheet model helps one carry out simple and fast analyses whenever they are needed. At the Seagate Technology's Recording Head Operations Wafer Manufacturing facility (Bloomington, MN) industrial engineers who worked on capacity planning devised their own versions of static spreadsheet models over the years. As useful as these individual models were, being highly custom-tailored and decentralized made them hard to cross-use and manage. To overcome this problem, the IE department designed and implemented a centralized spreadsheet based static capacity model with features that allow industrial engineers to create model outputs the way they want.

1 INTRODUCTION

Spreadsheet based capacity modeling has a very important role in the day-to-day functioning of many industrial engineering departments, particularly in the semiconductor and similar industries. Other types of fab capacity planning tools can be used for a variety of analyses but when it comes to quick answers for urgent questions/problems, spreadsheet models are what industrial engineers turn to most of the time.

The situation was no different at the Seagate Technology Wafer Manufacturing facility in Bloomington. Tasked with performing the IE duties for different fab areas (such as Photolithography, Electroplating, etc.), industrial engineers developed their own spreadsheet models. Over time, the IE department ended up with several different models where only one IE would know the details of one model. Although there was a common level of knowledge within the department, answering questions could be difficult in the absence of one industrial engineer. On the other hand, having several models created a decentralized structure for the modeling assumption set. Assumptions like equipment utilization, uptime, time allotted for engineering experiments, as well as time standards were kept in different places and merging this data together, when needed, was not easy. Also, shared data such as inline yield, product volume and schedule, were being entered more than once, creating a waste of time. All of the factors mentioned above started a discussion within the IE department where everybody agreed to optimize the static capacity modeling system.

2 METHODOLOGY

There were four phases of this study. Phase I involved the selection of the best possible static capacity modeling alternative. Once that was set, Phase II took place, where the main goal was to gather user requirements for the model. Phase III was about designing the data structure and information flow. And, in Phase IV, the new modeling tool was implemented, verified and validated.

2.1 Phase I

As far as alternatives, two major options stood out from the beginning: an internal versus an external model. An internal model would be developed within the company and would therefore be more customized. An external model would be an off-the-shelf software package that is proven to work. It was decided that a Kepner-Tregoe decision making study would be the best tool to use in this situation. Kepner-Tregoe is a procedure that helps the user find the best among many options by comparing user input. In this procedure, the user identifies the features that have to exist in all of the alternatives (the "must-have"s) and the features that are desired ("want"s). A "value" then needs to be assigned to every "want", defining the level of importance of that feature to the user. Once this framework is com-

plete, each alternative is checked to see if it satisfies the "must-have"s and gets a score for each of the "want"s. A final score is calculated based on this data and the alternative with the highest score that satisfies all the "must-have"s wins. (Kepner-Tregoe, Inc. is a corporation having a business address of P.O. Box 704 Princeton, New Jersey 08542. For more information on Kepner-Tregoe, visit http://www.kepner-tregoe.com)

After a series of discussions, 16 "must-have"s (Table 1) and 28 "want"s (Table 2) were identified.

Table 1: Static Model "Must-have"s

STATIC MODEL MUST HAVES	
Input	To enter multiple routes
Input	To enter a schedule by products in Seagate terms (DGR)
Input	To enter time standards by tool
Input	To enter load size assumptions by tool
Input	To enter uptime assumptions by tool
Input	To enter utilization assumptions by tool
Input	To enter rework assumptions by operation
Input	To allocate engineering time by tool
Input	To enter number of tools available in time (months, quarters)
Output	To retrieve capacity requirement statements by tool in time (months, quarters)
Output	To detail output in Seagate terms (by tool or DGR value)
Performance	To run 'what-if' scenarios
Performance	To run scenarios quickly (seconds/minutes)
Performance	To have route validation -- detail the missing operations
Use	To provide a back-up model developer/supporter
Use	To provide consistency between different sites

Table 2: Static Model "Want"s

STATIC MODEL WANTS		Value
Cost	Low cost to modify views/functions	7
Cost	Low cost to add views/functions	7
Cost	Low resource commitment for initial entries	7
Extendable	To use with queuing theory model	5
Extendable	To use with dynamic model (table feeding or direct input)	9
Extendable	To enter new equipment sets	10
Input	To enter fixed and variable time standards (range of load sizes)	9
Input	Ability to vary parameters in time	10
Input	Low level of manual entry	10
Input	To enter a buffer % by tool (upside)	3
Input	To enter cascading time standard matrix (First Wafer Effects)	9
Input	Tie-in to actual yields/reworks/Xsite data	8
Interface	To have cut/paste capability	10
Interface	Spreadsheet interface	10
Maintenance	Low level of maintenance	10
Output	To outline top constraint tools/DGR values/input variables by month/quarter/etc	10
Output	To analyze cross-qualification of tools	10
Output	Configures ouput to an OEE requirement by month	10
Output	Compares OEE requirement to OEE actuals by month	10
Output	Create graphs to communicate information	3
Output	To print results on "1 page"	3
Output	To customize view with colors	3
Performace	To open file (versions) from any PC	4
Performace	To have multiple users running scenarios at the same time	8
Test	Ability to troubleshoot formula/macro errors internally	9
Test	Documentation on troubleshooting and use	7
Use	Easy to select one tool's variables to analyze	8
Use	Easy to train new users	10

Among the "Must-have"s were:

- Ability to run "What-if" scenarios
- Ability to provide answers quickly (in seconds/ minutes)
- Ability to provide consistency between different manufacturing sites
- Ability to allow time-phased data output.

The following were some of the "want"s that were identified:

- Low level of maintenance
- Ease of access and use
- Compatibility with the dynamic simulation model
- Ability to debug internally.

As a result, both alternatives met the "must-have" criteria. When the values were assigned to "want"s and the final Kepner-Tregoe score was calculated, it was seen that the internal model alternative was the better choice and should be pursued (Table 3).

Table 3: Final Kepner-Tregoe Scores

KT SCORE	
INTERNAL	1031
EXTERNAL	726

2.2 Phase II

Once the internal model alternative was selected, the next phase involved designing the model that would satisfy as many of the user requirements as possible. In other words, the scope of the model needed to be identified. The exercise that was carried out for the Kepner-Tregoe study provided a good baseline for the user requirements, but for this phase a more detailed list was needed. Each industrial engineer came up with a list of features for the areas they are responsible for. Some features were crucial, some were valuable/timesaving, and some were "nice to have". After a series of meetings, a common list was created. The following sections outline some of the critical requirements put forward.

2.2.1 Model Flexibility

One of the most requested features was model flexibility. Flexibility in this case was defined as the ability to change system variables in the model rather than the assumptions database. Ability to change assumptions like tool qualification times and time allocated for engineering experiments in the model would certainly provide the IE with more capability. But model variables like time standards, utilization levels and number of tools are believed to have a higher importance and a higher chance of being analyzed. On the other hand, although more flexibility generally meant a computational inefficiency and longer run times for the model, a certain level of capability was needed to be attained. As a result of the flexibility discussion, approximately 80% of the assumptions were decided to be carried on to the model and the remaining 20% could only be changed in the assumptions database, if needed.

2.2.2 Time-Phased Parameters

Model inputs like product volume and yield are most of the time planned to change over time periods. Therefore there was little room for debate over these parameters. Assumptions like time standards, number of tools, tool uptime may or may not be time-phased based on preference. The advantage to having these inputs time-phased would be the ease of performing sensitivity and what-if analyses as more than one scenarios can be executed through one model run with the use of time-phased parameters. The disadvantage would again be increasing file sizes and longer run times. Based on the facts above it was decided that tool related assumptions (number of tools, availability, utilization, etc.) should be time-phased and inputs like time standards, tool qualification matrix and products should not.

2.2.3 Fixed/Variable Time Elements

One of the commonly faced problems of static capacity modeling is the changes in the equipment load size and the amount of work needed to update the time standards accordingly. In order to make it easier to update the new model, all the existing time standards were studied and the individual time elements were identified as "fixed" or "variable". A fixed time element was defined as the one where the total time to complete the step does not depend on the load size. A variable time element implied a dependency on the load size. For example, the time it takes to load the cassette into the tool would be fixed, as no matter how many wafers are in the cassette, it would still take the same amount of time to put the cassette in the tool. Similarly, placing wafers into the cassette would constitute a variable time element due to the fact that the time it takes depends on the number of wafers to be loaded in the cassette. After categorizing all the time elements in the fixed/variable format, it was possible to obtain the correct time standard by using Equation (1) as shown below. This method took away the need to revisit the time studies when there is a change in load size.

$$Run\ Time = (Fixed) + (Variable \times Load\ Size) \qquad (1)$$

2.3 Phase III

Having gathered and discussed all the user requirements, designing the data structure was next. Figure 1 depicts the architecture used for the model.

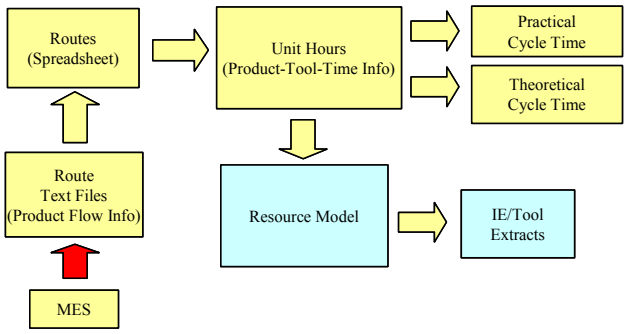

Figure 1: Model Architecture

2.3.1 Route Text Files/Routes

The modeling cycle started with the downloading of the process flows ("routes") from the Manufacturing Execution System (MES), which is the system used for tracking the lots in the wafer fab. The routes were downloaded in the text format (*.txt) initially. This was planned to be a manual process. Once all the routes were downloaded, a second step would take place where the individual routes would be combined into one spreadsheet file named "Routes" through the use of a macro.

2.3.2 Unit Hours File

The "Unit Hours" file was designed as the main database for the static capacity model. Product flow information was combined with equipment and time standards in a spreadsheet format. All assumption entries were to be kept here. The file was saved in a way that supported shared use, allowing multiple industrial engineers update the file at the same time. Features like exception reports, error checks, and product lists were built in to facilitate the management of the database. A backup of this file needed to be kept at all times due to the importance of the information it contained.

There were two useful reports that were by-products of this file:

- Theoretical Cycle Time Report: Calculates the theoretical cycle time for the products in the database using a load size of 1 for each step on the route
- Practical Cycle Time Report: Calculates the practical cycle time for the products in the database using the assumed load size for each step on the route.

2.3.3 Resource Model

The static capacity model was named "Resource Model" as it had the capability of calculating both equipment and labor requirements for the assumption set given. In order to build a model, a resource model generator macro was run which in turn gathered the assumptions from the Unit Hours file and created the necessary spreadsheets, entering the assumption values and the formulae in the spreadsheet. Essentially a new model could be created as many times as wanted just by running the generator. Once the generator finished running, the model was ready to use except for two pieces of information, product volume and yields. As soon as these were entered, the model was able to calculate equipment and labor requirements.

Some of the assumptions, like the product set to be modeled or the number of time periods, had default values to begin with, but during the execution of the resource model generator the user was asked to confirm these values. Also, as explained in the previous sections, some of the assumptions were not carried into the model from the Unit Hours file and therefore cannot be changed in the model.

Many different features were built into the model, such as the ability to create smaller size models for individual "Area IE" use or the capability of creating bottleneck lists.

2.4 Phase IV

After the design stages were over, implementation phase of the project took place where all the data analysis and entry were completed and the model was validated by comparing the model output to the fab metrics and previous static model outputs.

Initially, a sample data set was entered into the database for verification purposes and the debugging of the model was done on that set. After completion of verification, all assumptions including tool information and time standards were entered in the database. Finally, the model was run several times with different product sets and volumes in order to validate the model results.

Upon completion of the aforementioned tasks, a user manual was created, detailing the data structure, input/output schemes and the execution procedure. This manual was instrumental in cross-training efforts that were carried out later on.

3 CONCLUSION

This study stemmed from the need to have a centralized and capable static capacity model to replace the useful but hard to manage individual models. The methodology followed was straightforward and efficient where a systematic approach was used to choose the suitable alternative and user requirements were discussed in detail. This was deemed crucial as it was targeted that the new model would provide answers to all the common questions industrial engineers face every day. Design of the model also followed the same principle, supporting a lean and flexible data structure. Finally, the implementation was carried out without any major problems. The project took approximately 6 months and the static capacity model is in use now for over a year at Seagate Technology's Wafer Manufacturing facilities and has proved to be efficient and accurate. Although no formal study was conducted, the time saved by industrial engineers was significant. Elimination of the redundant tasks alone resulted in sizable timesavings. The turn-around time for what-if analyses also came down considerably. In addition to that, the new model provided a common base for different manufacturing sites of Seagate Technology improving congruency, communications and the overall confidence in the modeling process.

ACKNOWLEDGMENTS

The authors would like to thank Kimberly Jones, Gherense Neguse, Lee Phandanouvong, Juan Manuel Torres and Ummul Yamani for their help and cooperation.

AUTHOR BIOGRAPHIES

ORKUN OZTURK is a Senior Industrial Engineer at Seagate Technology's Wafer Manufacturing facility in Bloomington, Minnesota. He received a B.S. degree in Industrial Engineering from Bogazici University in Istanbul, Turkey and an M.S. degree in Industrial Engineering from University of Wisconsin-Madison.

MELISSA BOOM COBURN is the Industrial Engineering Manager at Seagate Technology's Wafer Manufacturing Facility in Bloomington, Minnesota. She holds a B.S. degree in Industrial Engineering from Iowa State University. Earlier in her career she worked as a Manufacturing Engineer, Industrial Engineer and Training Manager.

STEVE KITTERMAN is a Project Manager at the Longmont, Colorado Desktop/Mobile Product Design Center. He has a B.S. and an M.S. degree in Industrial Engineering from Purdue University in West Lafayette, Indiana. His work history includes Manufacturing Engineering, Industrial Engineering and Planning roles.