

SIMULATING SEMICONDUCTOR MANUFACTURING SYSTEMS : SUCSESSES, FAILURES, AND DEEP QUESTIONS

Karl G. Kempf

Manufacturing Systems
Intel Corporation
5000 W. Chandler Blvd.
Chandler, Arizona 85226, U.S.A.

ABSTRACT

Over the past several years, engineers at Intel Corporation have used modeling and simulation techniques to solve various high volume manufacturing problems. Our current goal is to use a persistent model over the entire life cycle of a factory to promote integration and continuous improvement of all of the components of the manufacturing system. The uses of this model are detailed as well as our progress towards realization of this goal. A summary of problems that we have encountered along the way is included as both a warning to those who have a similar goal and as a work list for vendors of simulation packages. Two disturbing questions about the basis of manufacturing simulation are asked that should be of concern to practicing simulation engineers as well as university researchers.

1 INTRODUCTION

Whether the domain is governmental or commercial or educational, whether the focus is business or technical or organizational, whether the formulation is spreadsheet or linear programming or discrete event, simulation is a powerful tool for understanding and enhancing the performance of complex man/machine systems. In countries where the Gross National Product is based largely on the production of goods for sale, understanding and enhancing the performance of complex manufacturing systems is particularly important. In this application, simulation in its many forms can have an impact measured in many percents of corporate productivity, market share, and profitability. In the extreme, global competitiveness can be positively influenced.

At Intel Corporation, simulation plays an indispensable role. Critical business questions are studied from various perspectives with a host of simulation tools. Product designers, seeking to realize ever more useful and reliable chips, employ simulation

to study the behavior of everything from individual transistors to full computer architectures. Process designers, striving for ways to build ever smaller and lower power devices, use simulation to study everything from the behavior of plasmas to the interaction of individual steps in long manufacturing processes. But the focus of this paper is manufacturing systems where the ephemeral business plans, product designs, and process specifications come together and are realized in actual products. In manufacturing, Intel betters the ancient philosophers dream of turning lead into gold. We turn sand into computers.

For manufacturing engineers, simulation in its various forms has application in each life cycle phase of a semiconductor factory including 1) design, 2) production ramp, 3) early high volume production (when the market could absorb more units than can be made with the new capacity), and 4) late commodity production (when old capacity can supply more units than the market desires). This is true whether the factory under discussion performs device fabrication, component packaging, or circuit board assembly. In the next section on Integration, the goal of our simulation efforts will be described along with a report on our progress toward that goal. The following section on Practical Difficulties will make available for others with similar goals the pitfalls to avoid and problems that vendors of commercial simulation packages need to help solve. The final section on Theoretical Difficulties will briefly describe issues that we have encountered in the march towards our goal that question the foundation assumptions of simulation and should stimulate the imaginations of university researchers.

2 INTEGRATION THROUGH SIMULATION

The cost of a fully outfitted semiconductor facility has reached two and a half billion US dollars, with a potential revenue stream of more than twice that per year. At this level of investment and return, nothing can

be left to chance. The decisions needed to design, ramp, and operate the manufacturing system must be data as well as experience driven. All of the components of the system - from layout to lot size, from decision policies to automation systems - must be well integrated during every stage of the factory life cycle. Continuous improvement of every component must be supported across the life cycle. Our vision is to achieve integration and continuous improvement by using a persistent simulation model throughout the life of the factory. This model would be used by a variety of simulation tools for solving a variety of problems. We have been focused for a number of years on the process of realizing this vision.

2.1 Simulation for Manufacturing System Design

The first step in the process is to design the high volume manufacturing (HVM) system for "manufacturability". Even before excavation for the building begins, the factory exists as a fully functioning entity in simulation. To build the basic simulation, a number of things must be known, including :

- the business goals of the factory,
- the flow of operations in the manufacturing process,
- the types of processing equipment to be used, and
- the level and types of automation to be included.

Given the basic simulation model, verified as well as possible in the absence of the working factory, a long sequence of iterative studies begin. The sequence is long because there are a variety of stakeholders who must agree and there are numerous factors that must be considered. The sequence is iterative since few if any of the factors are independent.

The number of each type of equipment required to satisfy the business goals is determined (capacity), complicated by the integer nature of equipment and the uncertainty in run rates and availabilities. The floor execution policies for material release, management of work in progress (WIP), equipment maintenance, cross training, and engineering improvements are designed and tested for their positive impact on business goals. Transfer sizes and equipment layouts are evaluated and used to begin to scope the automation system. Consideration of safety, ergonomics, and productivity are factored into the automation design and the staffing levels. Construction constraints including local building codes and federal environmental rules are applied. Since execution policies modify capacity, construction constraints complicate layout optimization, automation impacts policies, and staffing impacts business goals - to mention a very few of the interactions - the iteration proceeds through many cycles. But the hard won result, the HVM system design with tradeoffs quantified, is useful over the life of the factory.

2.2 Simulation for the Production Ramp

With the high volume manufacturing system design as a goal, and a completed building available at some estimated point in the future, the second step in the process is to choreograph the ramp from zero production to full production in the shortest time and most organized manner possible. This generally means getting one instance of each equipment type running so that initial production material can traverse the whole process, then adding further pieces of each equipment type to smoothly ramp capacity up to the designed maximum. At least three factors come into play, and all can be studied with the help of the HVM model developed during the design phase.

One factor is the realistic delivery schedule that can be achieved by the various suppliers of equipment. Early delivery ties up capital unnecessarily. Late delivery delays the critical production ramp. Meeting a tight delivery schedule may strain the supplier beyond what is physically possible. Compromises are required to get the right equipment delivered at the right time.

Another factor is the rate at which skilled personnel can be available to install the equipment on the production floor and to qualify it to accurately execute the process specification. Since each equipment type involved requires a different skill mix, having the right skills available at the right time over the whole ramp is a challenge.

The third factor is the rate at which competent personnel can be made available to operate the installed and qualified production equipment. This rate can be constrained by the availability of local people to join the company, the ability of these people to acquire the operational competencies, or the availability of trainers to facilitate the training.

Since each of these three rates has multiple components, and there can be uncertainty and fluctuation in each component, simulation is a useful approach to determining what the constraint to ramping will be under a variety of scenarios before the ramp actually starts. And of course, as surprises occur during the actual ramp, the same simulation can be used to evaluate their impact, plot work-arounds where possible, and to re-plan when necessary.

2.3 Simulation for Early Production

The high volume manufacturing design may be completed many months before the ramp begins, and the production ramp may take many more months to realize. This is more than enough time for conditions in the market place to change causing modification of the business goals of the factory. And of course, any number

of difficulties may have precluded the precise execution of the plan for the ramp yielding a factory that is different in some ways from the one that was initially designed. Each of these changes and deviations must be identified and addressed.

An excellent starting point for optimizing early production, and one that can be used well before the ramp is completed, is the model built during the HVM design phase. Since the model is maintained and used throughout the ramp, it is available to evaluate HVM solutions to changed business conditions and modified ramp execution. Before the end of the ramp, it may be possible to order and install more or less equipment than originally planned. After the end of the ramp, policies can be modified and personnel can be re-trained. The impact of each of these changes on early production can be quantified through simulation using the HVM model.

In fact, even if everything is just as expected at the end of the ramp, and the product issuing from the factory is successful in the market place, the HVM model will continue to be used in at least two ways through the early production phase.

On one hand, as production matures over the first few years of factory operations, execution policies will need to be continuously improved to satisfy the dynamics of the market and the pressure that manufacturing is always under to become more efficient. Again, the continuously updated and validated HVM model is the tool of choice to support evaluation of policy changes and interactions between policy changes prior to implementation on the production floor.

On the other hand, automation tools will have been built into the shop floor control system to support the execution of the material release, WIP management, equipment maintenance, and engineering improvement policies. The HVM simulation model will be used at the core of these tools for automated policy execution since it will have been available early in the development cycle of the automation system and will always be the most complete, correct, and consistent model obtainable. Policies developed by off-line simulation using this model will be communicated to on-line automation systems using this model to direct policy execution.

2.4 Simulation for Late Production

The ramp and early production scenario just described assumed that the market could absorb all of the product that the factory could produce, and so a major part of the business goal was to maximize the output. Later in the factory life cycle when the product for which the

factory was initially built becomes a commodity, the business goals change considerably. The capacity is mostly amortized. A few more processes have probably been added to the factory along with a number of other products. The factory is now selling throughput time (TPT) - the ability to offer and deliver short lead times for all of its products. And the factory is very cost sensitive.

Running multiple processes and multiple products over one equipment set can be very complicated. All of the executional policies that were vital to maximizing output previously now become vital to minimizing TPT and cost. The material release and WIP management policies must evolve to handle low volumes of a wide diversity of products. Engineering improvement policies focus more on time saving and cost cutting projects as do the equipment maintenance policies. Staffing and cross training policies are now focused on headcount reduction and productivity improvement even more than previously.

Once again the HVM model that has been maintained and updated since the initial design of the factory can be used in a variety of ways. The model can continue to support the modification and improvement of executional policies to satisfy the new business goals. The model along with the improved policies can continue to support floor execution through the automation system. And, of course, the model is still the best capacity estimator for the factory, providing sanity checks for market-driven changes in product mixes and volumes experienced frequently in a commodity factory.

Finally, the HVM model is invaluable for the last ramp of the factory - ramp down and closure. Ramping down in an orderly fashion is at least as hard as ramping up, although the problem set to be considered is different. There is much choreography to be done to de-install machines, and to re-train and re-deploy personnel. And it must all be done without the adrenaline of the ramp up. But it may be just as economically important to the company to have a clean ramp down as it was to have an aggressive ramp up, and it is probably even more important to the psyche of the manufacturing organization.

2.5 Intel Today and Tomorrow

The vision described here is based on a single continuously updated simulation model that is persistent over the life cycle of the factory. It is key to designing and ramping the high volume manufacturing system since it is one of the only sources of reliable data to drive critical decision making processes. It is vital to early and late production phases since it contains most of the details required for continuous improvement and

automated execution of the operational policies that drive floor execution. In all these uses, it is superior to any documentation since it is a computer executable specification not open to different interpretation by different personnel. Given the difference between this vision and conventional manufacturing practice, and given the scope of this vision, it is being implemented at Intel in piecewise fashion with an opportunistic approach.

It has been common practice for many years among industrial and manufacturing engineers in our fabrication and assembly/test facilities to build simulation models to answer capacity and staffing questions, with spreadsheets giving way to discrete event tools. These models have been built earlier and earlier in the factory life cycle, resulting in the current practice of developing the model well before groundbreaking to support the earliest capital purchase and personnel hiring forecasts. Unfortunately, once these early questions have been answered, the models tend not to be maintained. We are striving to use this experience to launch the concept of the design phase model that persists (Hilton et al. 1996, Sohn et al. 1996).

Over the past few years in operating factories, we have begun to simulate operational policies, especially material release and WIP management policies, before sending them to the shop floor for manual execution. It has been the case that new models are built and validated to pursue this work. Since continuous improvement of the policies is required, a few of these models have been updated and maintained their usefulness over several years. We are striving to modify this current practice to promote maintaining the design phase model instead of building a new model.

Although our manufacturing automation group has used simulation to design material handling systems for some time, only recently has it begun to develop and deploy model-based manufacturing planning and scheduling systems. A few assembly/test lines now use a simulation-driven planning system that regularly passes scheduling data to the shop floor control system, with proliferation through all assembly/test factories underway. Similar systems are in development for our fabrication factories. These systems support off-line policy development linked directly to on-line policy execution, and force models to be maintained by appropriate factory personnel. This new direction in our automation group fully supports our vision of the persistent model for design and execution of HVM.

Our most recent advance has been to work toward deciding important HVM questions while fabrication and assembly/test processes are still under development, and of course models and simulations are the tools that

have been used. This promises to much improve both the result and the efficiency of the HVM design process that previously was strongly driven by experience and opinion rather than data. It also facilitates the use of existing simulation tools to control ramping since the detailed HVM model will exist at the appropriate time in the hands of the appropriate people.

Although we have followed a jagged trajectory, we will be in position to realize the full vision within the next few years in both fabrication and assembly/test factories. There will be HVM design processes in place that require a model to be built well before building the factory, and require the model be maintained for ramping. There will be automation tools in place that will use the model to continuously improve and automatically execute shop floor policies through all production phases. We expect the results of our integrated efforts to be a step forward in the theory and practice of manufacturing system science.

3 PRACTICAL DIFFICULTIES

Neither our progress so far nor our efforts over the next few years are without practical difficulties. Almost all of these difficulties will be encountered by anyone interested in the modeling and simulation of manufacturing systems. The solution of most of the difficulties will come from advances in modeling and simulation software tools. Other difficulties will be solved by improved training of simulation and manufacturing engineers. The process that we have developed over the years for modeling and simulation has only a few phases, but it can be used to motivate the description of the practical difficulties we have suffered.

3.1 Designing the Model

At Intel, the model design phase centers around the stakeholders, the HVM personnel who are the customers of the modeling and simulation effort. It is first necessary to collect all of the questions that the stakeholders desire to have answered. These questions drive the “what” and the “how” of the modeling effort. We are striving to design the simplest model that can answer all of the questions. The “what” is deciding the entities and behaviors that need to be included. The “how” has to do with the consistent level of abstraction at which to represent. Too many inclusions with too much detail adds to development and run time. Too little of either leads to poor or wrong answers. This initial model design effort will clarify assumptions, including ones about the availability and accuracy of data, all of which need to be checked with the stakeholders.

In our experience, it is vitally important to include the plan for model validation and the planned set of experiments (along with metrics) in the design discussions with the stakeholders. This calibrates the stakeholders who do not always know much about modeling and simulation, and calibrates the modeling and simulation personnel who are not always experienced in manufacturing. It also builds teamwork between the stakeholders and the modelers for the hard work that lies ahead.

Only at the end of the design phase can one decide whether the modeling and simulation project should go ahead, should be delayed, or should terminate. In hindsight, it is clear that we have had projects that should have been terminated, or at least delayed at this point because the questions of the stakeholders could not be answered by modeling and simulation, or the appropriate data was not available to the required accuracy, or some other related overwhelming difficulty. This is indeed a tough call, but pressing ahead in these circumstance only delays and increases the pain. It is better to take the pain early. It is unfortunate that this very important phase of design and negotiation is rarely included in the formal education of modeling and simulation personnel in either university or vendor training.

3.2 Building the Model

Another decision that should be made after the model design has been completed is tool selection. One is trying to select the simplest tool that can be used to answer all of the questions. Unfortunately, we have often made mistakes here. We have certainly taken problems that could have been solved adequately with a spreadsheet and applied a discrete event simulator, with much wasted time and effort. And we have used a spreadsheet on problems that should have been answered with a discrete event simulator, with poor answers as a result. The error has always been deciding the tool before designing the model, usually based on the convenience or preference of personnel on the modeling team. Better to design the model, then select the tool, even if it means changing team personnel.

There is an obvious practical problem here. Ideally, the team would have a suite of tools available to it, and be trained in the entire suite. Unfortunately, given the cost and complexity of many commercial tools, the best that might be achieved practically is the availability of two or three tools, with only one or two team members trained in each. This is origin of many poor model-to-tool mappings, and subsequent difficulties.

Even having selected the most appropriate tool, modeling building has frequently been difficult for us.

With the best intentions, manufacturing organizations do not always collect all of the data required for the model. They frequently have great difficulty describing floor procedures to the model engineer in adequate detail. And facilities for automated data transfer from shop floor data systems to modeling and simulation tools are rare indeed. For these and other reasons, model building is a laborious manual process that almost always over-runs its schedule. Again, the formal education of simulation engineers is usually lacking in overcoming any of these difficulties. In addition, tools from simulation vendors are too often weak in this area.

Selecting the best tool and building the model gets us to the most difficult problem in this phase, that is model validation where the model is tested to see if it gives "correct" answers. When validating against an existing factory, historical scenarios can be used. Since all of the details of everything that happens on the factory floor have not been (and in fact, never could have been) deposited in the model, deciding what is correct is not a trivial matter. When validating a model for a non-existent facility, a wide variety of simple tests must be done. In either case, thinking through the validation approach and seeking stakeholder agreement to the approach during the previous design phase is critical because stakeholders have intuition. Four results are possible, and we have experienced all of them.

a) The answer produced by the tool is in agreement with the intuition of the stakeholder, and the answer is correct. This is a happy circumstance. We have more than our intuition to justify our actions.

b) The answer produced by the tool is in agreement with the intuition of the stakeholder, and the answer is wrong. This is the worst possibility. We will charge off in the wrong direction using the results from the tool as a shield against all adversity. It is very difficult to dissuade someone whose intuition is in agreement with a (wrong) simulation result. The mistake here is usually that we built the model including all of the elements that went into the intuition, and so the tool produces results in line with the intuition (as you might expect). The problem is detecting that the answer produced by the tool is wrong.

c) The answer produced by the tool is not in agreement with the intuition of the stakeholder, and the answer is wrong. This is what we see in testing the tool as we develop and debug it. We sometimes use a suite of increasingly difficult test problems about which we have sound intuition. The worst that can happen here is that the stakeholder sees the tool before we have removed all the bugs, and loses confidence in the tool, perhaps undeservedly.

d) The answer produced by the tool is not in agreement with the intuition of the stakeholder, and the

answer is correct. This is the most important case. If the tool disagrees with the intuition of the stakeholder, the tool is wrong from the stakeholder's viewpoint. However, if the tool is in fact correct, the unexpected answer is very important. The only way to make the correct answer stick in the face of disagreement with the stakeholder is to have the tool very carefully validated, and to have had the stakeholder buy into the validation process before the disagreement surfaces.

Better analysis tools would make validation much easier, and would also be very useful in the last step in this phase, executing the experiments planned during the design phase. The task of the simulation engineer, from experiment to experiment (and validation to validation) is to track down every performance change in the simulated system. Doing this using the analysis tools available in most commercial simulation packages is error prone and time consuming at best. And, of course, once a change has been detected, explaining its origin is the reason for doing the experiments.

Missing changes and mis-explaining noticed changes are the major errors made during validation and experimentation. The reason that automated analysis tools would be so useful is that experience shows our intuition to be rather lacking when dealing with small changes in complex manufacturing systems. This holds for both the stakeholders who know the real manufacturing system and the simulation engineers who know the model and the tool.

3.3 Maintaining the Model

Given our goal of using the model through all of the life cycle of the factory, the first two phases described here are only the beginning. The design, construction, validation, and initial experiments to define HVM are important, but are really only the infancy of the model. Maintaining the model through the initial production ramp then through early and late volume production is where the majority of effort will be spent.

Not many new problems arise during maintenance of the model. The model must be updated with all of the same problems described above for initially building the model. Our experience indicates that a significant number of things change in a working factory that are a concern on a daily or weekly basis. The model must be re-validated, if not at every update, then at least at every few updates or after a defined time period has passed. All of the validation problems listed above are still encountered here, and must be addressed to keep the model from losing the confidence of the users, both stakeholders and modelers. Since the model is there to supply data upon which to base operational decisions, for both off-line continuous improvement and on-line

execution, all of the problems associated with doing experiments recur.

The difficulty that we have seen in this phase is discipline. The engineers chartered to keep the model vital frequently have a variety of other assignments and responsibilities. Maintaining a complex model requires a lot of time, and underestimation of that time, or deprioritization of that time to instead address urgent tactical problems can lead to situations where the model is required, but is stale. Then a big effort must be launched, and yet another mechanism of loss of confidence in the model can be accidentally activated. The best answer is simply to develop the discipline to give the model regular maintenance, even when its next use is not clearly in sight.

4 THEORETICAL DIFFICULTIES

In all of the work described so far, at least two high level assumptions have been made implicitly. One is that we are basically modeling the appropriate entities to answer the questions that our stakeholders need answered. The other is that the manufacturing systems that we are modeling are basically well behaved. From our work, we have strong indications that both of these assumptions need to be very closely examined. The deep questions that we have been asking ourselves are outlined here along with pointers to our published work on these topics for those who wish to delve further.

4.1 What Should be Modeled

On the manufacturing floor of a typical factory, one comes into contact with two distinct types of entities. On one hand are entities that have no inherent intelligence. They are dumb. This includes machines, jobs, tools, buffers, and transporters as examples. These entities have status that is tracked since decisions must be made about how they should be used on a minute to minute basis to achieve the performance goals of the factory. But these dumb entities can not make allocation decisions themselves. On the other hand are entities that have high inherent intelligence. They are smart. This includes machine operators, maintenance technicians, area supervisors, and shift managers as examples. These entities can make decisions about the use of the dumb entities, and employ status information to do so, including decisions about WIP movement, capacity allocation, scheduled and unscheduled equipment maintenance, batching, setups, and other related activities. In our factories, one goal of the automation system is to supply status and support decision making.

The point is that both the dumb entities and the smart entities are important to the performance of a

manufacturing system. It is clear that the capacity and speed of a factory depend heavily on the capabilities of the equipment set and that variability in factory performance is strongly tied to pseudo-random equipment breakdowns. However, it is equally clear that speed and capacity is directly impacted by the quality of decision making by the smart entities, and that random behavior in the smart agents also shows up as factory variability. In fact, in factories with which we have been associated, the “equipment problems” usually pale by comparison to the “people problems” in terms of achieving and sustaining uniformly high system performance.

Furthermore, to understand and improve manufacturing systems, we need tools to study the dumb entities, the smart entities, and the interactions between the two. Unfortunately, most modeling and simulation tools that are available for the study of manufacturing systems are equipment-centered. Factories can be simulated in great detail without any consideration of the human element. The quantities, properties, and arrangement of the equipment can be altered to play out a wide variety of “what-if” scenarios. The equipment is given decision making capability though dispatching rules that are executed instantaneously and uniformly throughout the simulation run. When operators and technicians are included, the equipment retains the capacity to make decisions while the actual decision makers are treated as tools. A machine decides what to run next, then checks the current simulation data base to see if an appropriately qualified operator is available to load the chosen job.

We propose that new and important insights would be gained if dumb entities and smart entities were modeled and simulated with equal fidelity. We desire the ability to play a wide variety of “what-if” scenarios around the smart entities too, and to include the interactions between the different types of entities in the simulation. We speculate that a significant fraction of the complex global behavior observed in a semiconductor manufacturing system emerges from the interactions between the smart entities, and between smart and dumb entities.

A model has been constructed and a new simulator implemented to test these ideas. The full results have been published elsewhere (Spier and Kempf 1995), and only one experiment is described here to pique the reader’s interest. Suffice it to say that the factory is a simple one containing a manufacturing process with six steps, an equipment set with five machines, and a staff of three floor personnel (two operators, one maintenance technician). The experimental data presented in Table 1 was generated by varying the characteristics of the smart objects, an impossibility in most other systems.

The rows marked “NON-COOP” and “COOPERATIVE” test the factory performance of two methods by which the three smart objects interact. In “NON-COOP” mode, the tactic for breaks is simply to take them whenever there is no machine requesting service and it has been long enough since the last break. The advanced decision procedure considers when preventive maintenance is due and whether other smart objects are off the manufacturing floor. For example, operators will not go on breaks together, will service the machines of the operator who is on break in addition to their own machines, and will try to take breaks when machines are being maintained. This is called “COOPERATIVE” mode.

The rows marked “LAZY”, “REACTIVE”, and “PROACTIVE” test the factory performance of three methods by which the smart objects function individually. The default mode for executing WIP management policies is called “REACTIVE” mode. The smart object simply waits for a machine request, services that request as dictated by the current decision policy, then waits for the next request. There is also “PROACTIVE” mode in which the smart object, upon finishing a request, uses the information available to determine the next machine which will make a request and moves into position before the request is made. Finally, there is “LAZY” mode, where, with multiple requests for service, the smart object selects to service the machine that is closest to its current position.

Table 1 - The Impact of Including Personnel

<u>PRIMARY</u> <u>CHARACTERISTIC</u>	<u>secondary</u> <u>characteristic</u>	<u>TPT</u>	<u>OUT</u>
NON-COOP	reactive	3837	780
COOPERATIVE	reactive	3605	840
LAZY	non-coop	4303	700
REACTIVE	non-coop	3837	780
PROACTIVE	non-coop	3818	793
COOP/PROACTIVE	-	3477	862

The row marked “COOP/PROACTIVE” shows the performance of the best of both approaches. Notice that the performance of the manufacturing system changes dramatically with a constant process and equipment model, but varying character and interaction of the smart objects that are normally excluded from consideration. Notice that the differences between the best (lowest) and worst throughput times and the best (highest) and worst factory output approach 20%. We

propose that this might be the magnitude of error in answering questions about a real factory using an equipment-centered simulation model.

4.2 What can be Simulated

In an informal manner, the word “chaos” frequently comes to the lips of engineers trying to operate manufacturing facilities. From a formal perspective, the discovery and application of the theory of deterministic chaos to natural systems has revolutionized work in many branches of physics, chemistry, and biology. We have been interested in applying the formal theory of chaotic behavior to begin to quantify the informal intuition of manufacturing engineers. The goal of this work has been to provide answers to three questions. 1) Can chaotic behavior be found in simulations of simple manufacturing systems? 2) If so, can chaotic behavior be observed in actual manufacturing systems? 3) If so, how can this behavior be utilized, minimized, or avoided? The experiments we have reported previously begin to investigate the first question.

There have been various attempts to define chaotic behavior in the literature, but no general agreement has been reached in this relatively young science. A component common to every definition reflects the chaotic system characteristic that small changes lead to large effects. This is in direct conflict with the doctrine of Newtonian determinism and its small perturbation assumption about natural systems. Such determinism concerns prediction of the future course of a system given a set of natural laws, the structure of the system, and a set of initial conditions. The Newtonian doctrine states that small approximations in the natural laws, small inaccuracies in the structural description, or small errors in the initial conditions could have only small effects on the resulting predictions.

Chaotic systems behave quite differently. Very small perturbations in structures, laws, or conditions produce very large changes in performance predictions as an inescapable consequence of the intertwining of small scale and large scale phenomena. Simple deterministic systems can exhibit chaos as orderly disorder. Their behavior is bounded, but in some important time scales unpredictable.

The manufacturing system that we have studied by deterministic simulation has only four processing steps and four pieces of equipment. But it includes one of the most distinctive characteristics of semiconductor fabrication, re-entrant flow (where every job comes back again and again to the same equipment type in completing the full process flow). Two metrics have been used to evaluate the performance of the modeled system. One is the distribution of TPT over all jobs,

where TPT is defined as the time elapsed between a job starting execution at the first step in the process to finishing execution of the last step in the process (in units of minutes). The distribution is summarized as a minimum, an average, a maximum, and a standard deviation. The other metric is the interdeparture time of finished jobs leaving the last step in the process flow. Exit times are recorded (in units of minutes), the sequential deltas between them computed, and basic patterns of interdeparture times identified. The patterns are named with an integer stating their length, and a character indicating the details of their sequence.

Small changes in this system fall into two categories. We have explored changes to policies for the release of raw materials and for the withdrawal of jobs from queues of work in progress, and have investigated changes to the contents and ordering of initial and dynamic queues. Large changes to TPT distributions and temporal patterns of finished jobs flowing out of the system have resulted.

Having shown that small policy or queue changes induce large performance changes, we altered the structure of the manufacturing system to find the origins of the chaotic behavior. This includes the mapping of processing steps onto production machines, the volume of work being added to the system relative to its capacity, and the size of batches being placed into batchable machines. It has been shown that each of these factors contributes to the complexity required for the onset of chaotic behavior.

Although the full results have been published elsewhere (Beaumariage and Kempf 1994), one experiment is described here to stimulate the interest of the reader. The experiment revolves around process Step-1 and process Step-3 in the model. These two steps run on the same machine type, and so there is one queue for work in process at this machine type which holds jobs waiting to run Step-1 and jobs waiting to run Step-3. The machine type involved batches three jobs at a time that must all be waiting for the same processing step (1 or 3). Batches are made up by finding the first three like jobs starting at the head of the queue. The experimental data presented in Table 2 was generated to demonstrate that very small changes in the dynamic contents of a queue precipitates very large changes in the overall performance of this simple manufacturing system.

The row marked “STANDARD” represents the simulation run where the queue was not dynamically altered. The row marked “SUB” shows data from an experiment in which a job waiting for Step 3 was removed from the tail of the queue at an arbitrary time during the run. In the row marked “ADD”, data is presented from an experiment in which a job waiting for

Step 3 is added to the tail of the queue at the same arbitrary time as in "SUB". The data in the row marked "SUB:ADD" is from an experiment in which a job waiting for Step 3 is removed from the head of the queue one second before a batch of Step 3s would have been loaded, held for two seconds, then placed back into the tail of the queue one second after a batch of Step 1s was loaded. This is equivalent in the simulation to swapping the order in which two batches were done at one particular point in a long simulation. In the base case, a batch of Step 3s were processed first followed by a batch of Step 1s. In the "SUB:ADD" case, the reverse sequence was forced.

Table 2 : The Impact of Small Queue Changes

	<u>TPT</u> <u>min</u>	<u>TPT</u> <u>ave</u>	<u>TPT</u> <u>max</u>	<u>TPT</u> <u>std dev</u>	<u>Pattern</u>
STD	567	739.3	922	61.8	33-A
SUB	552	817.8	1109	157.9	33-C
ADD	532	788.1	1195	153.1	165-A
SUB:ADD	645	858.0	1146	123.6	132-B

Three very minor queue perturbations produce very large differences in performance results. If unperturbed, all four runs would have shown the same results. But after minor perturbations midway through runs of a year of simulated time, the resulting TPT distributions are each very different than the base system, and are different than each other. Furthermore, all three resulting patterns are different from the base system and very different from each other. We propose that this indicates that manufacturing systems may not be as well-behaved as we assume. The possibility of chaotic behavior should be considered each time a simulation produces a surprise result.

5 CONCLUSIONS

Our work over the past several years is moving us closer and closer to applying modeling and simulation techniques during every life cycle phase of all of our manufacturing systems. We believe that the value of a model that persists over the life of a factory is great in the context of supporting both integration of all components of the systems and continuous improvement of the components and the system.

Unfortunately, there are many hurdles in the path of achieving this goal. They include issues in designing and gaining consensus for the model, building and validating the model, and using and maintaining the model over the life of the factory. Better education and improved tools are required to overcome these difficulties.

But even then, two questions remain that must be addressed further. One deals with what we include in our models, specifically how we include the human decision processes that drive the manufacturing system. The other deals with what we can expect from any of our various simulation techniques given the possibility of chaotic behavior in manufacturing systems. These questions require much more research to settle.

ACKNOWLEDGMENTS

This work has been supported by too broad a set of Intel manufacturing personnel to exhaustively acknowledge here. However, special thanks go to Ken Thompson, Don Rose, and Gene Meieran for management support and to Stephen P. Smith, Naiping Keng, and Vats Srivatsan for technical collaboration.

REFERENCES

- Beaumariage, T., and K. Kempf. 1994. The Nature and Origin of Chaos in Manufacturing Systems. In *Proc. SEMI/IEEE Advanced Semiconductor Manufacturing Conference*, 169-174. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.
- Hilton, C. M., G. Mazenko, L. Solomon, and K. Kempf. 1996. Assembly Floor Layout and Operation. In *Proc. Inter. Symp. on Semiconductor Manufacturing*, to appear. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.
- Sohn, B., A. L. Romero, and S. Mouck. 1996. Designing Fabs for High Performance. In *Proc. Inter. Symp. on Semiconductor Manufacturing*, to appear. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.
- Spier, J., and K. Kempf. 1995. Simulation of Emergent Behavior in Manufacturing Systems. In *Proc. SEMI/IEEE Advanced Semiconductor Manufacturing Conference*, 90-94. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

AUTHOR BIOGRAPHY

KARL G. KEMPF is Principal Scientist for Manufacturing Systems at Intel Corporation in Chandler, Arizona. He is interested in applying simulation to understand and enhance the performance of complex man/machine systems. He has practiced various forms of modeling and simulation in Gran Prix motor racing at Ferrari S.p.A., in cinematic special effects at Pinewood Movie Studios, at McDonnell Douglas in Space Station design, and at Intel Corporation in manufacturing automation.