APPLICATIONS OF ARTIFICIAL INTELLIGENCE IN FACTORY MANAGEMENT

Eugene S. Meieran
Karl Kempf

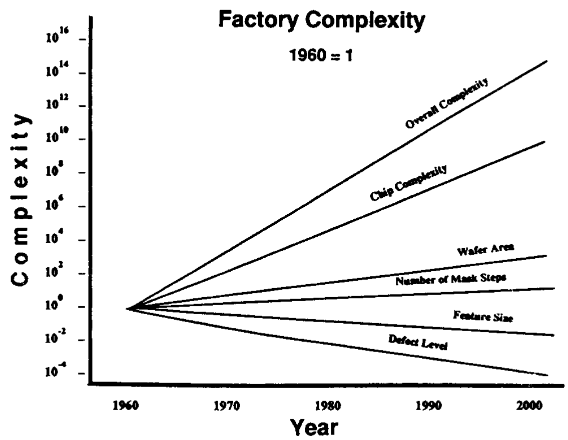
Intel Corp.
Knowledge Applications Laboratory
Factory Automation Department
3065 Bowers Ave., MS SC2-11
Santa Clara, CA 95051

Abstract

The management a factory manufacturing semiconductor devices has always been a difficult task, and is becoming even more challenging as the industry tries to respond to the influence of a number of concurrent powerful forces. For example, the relentless increase in process and product complexity continues to challenge technology capabilities, several emerging marketing forces have put new pressure on reducing product cost and improving product quality, and at the same time, new technology capabilities are creating new manufacturing and business opportunities at an ever-increasing rate. Because of these pressures on factory management, many companies are evaluating Artificial Intelligence (AI) technology to complement conventional computer technology in managing the manufacturing enterprise. Particular emphasis is being paid to evaluation of the use of experts systems for many domain-specific applications, and to use of knowledge-based systems for larger applications such as factory scheduling. When AI technology is applied to factory management, two issues are introduced that are not very important when conventional computer technology is used. First, the AI program has to be linked to the factory management or control system that contains the relevant shop floor data; this is usually not a trivial task. Second, introduction of AI technology to a factory floor is perceived as "new", even in a factory environment where the introduction of "new" processing equipment or procedures is routine, and non-routine "newness" on a factory floor is ALWAYS received with a healthy dose of skepticism. In this paper, these aspects of introducing AI-based technology to the manufacturing and administrative tasks encountered on the factory floor are discussed.

Introduction

As is well known by anyone involved in the semiconductor industry, we all live on the edge of a seemingly bottomless abyss, due to the ever-present pressure to competitively manage and sell ever-increasing device complexity. Figure 1 gives a view of the combined effects of technology complexity increases since the inception of the industry just about 30 years ago. The various aspects of complexity are ALL ultimately focused on the factory floor, which has to deal not only with the conventional view of complexity as the number of active elements present on a single chip, but with larger wafers, more process steps per wafer, a larger pin count, lower defect levels, better particulate control, etc. From a factory perspective, complexity has increased perhaps *15 orders of magnitude* relative to 1960, when the first wafers (containing a few dozen one-transistor chips) were manufactured. The issue is, how can one continue to manage these concurrent and interrelated increases in complexity? Although the problems associated with continuing every individual trend shown in Fig. 1 are formidable, there is no indication that any trend is starting to taper off. Two-hundred millimeter diameter wafers containing many hundreds of sub-micron geometry, multi-million cell chips, are a certainty in the next few years.



While conventional computer technology is now regarded as essential in manufacturing semiconductor chips and devices, Artificial Intelligence (AI) technology is now receiving attention from manufacturing engineers who wish to exploit its capability to handle computationally difficult manufacturing problems. AI is already being used by many companies to assist in the diagnosis and repair of processing equipment, to configure complex systems for customers, and for many other applications where human expertise is scarce, or where there is need to solve difficult management problems normally thought to require human reasoning capabilities.

One important management task that is receiving much current attention is that of scheduling; the allocation of the resources in a factory in order to provide product output required to satisfy business demands. In the past, many conventional software programs have been applied to the scheduling problem without a lot of success; recently, use of AI technology has been looked at as a possible alternative.

During the course of developing an AI-based expert systems as well as knowledge-based systems such as an AI scheduler, a number of non-technical issues arose, independent of the AI technology required to solve the problems. These issues can be divided into three categories. For illustrative purposes, scheduling will be used as the main application example:

1. Integrating the AI program with the conventional technology needed for scheduling a factory

2. Support and training for AI applications, software and hardware in a production environment

3. Getting over the hurdle of introducing *change* in a production environment; making AI acceptable to factory managers

In many ways, the last item is the most difficult to manage; there is an inertia or natural resistance to change in factory management as well as in physics. This is due, for example, to the tendency for *any* change to be perceived as perhaps strategically necessary but tactically difficult or impossible to fit in, from fear of doing something that upsets the production apple cart, from a lack of understanding of the long-term benefits of the change, from a lack of confidence in being able to deal with new technologies on the part of factory staff, etc. Dealing with this issue is certainly as important as dealing with the technology, and is much more critical to do properly.

Each of these three topics will be briefly discussed below.

Scheduling

Semiconductor factory scheduling may be characterized by three different sets of "complexity" issues, regardless of the nature of the computer technology applied to generate a factory schedule.

1. Logistic complexity

First, most semiconductor wafer fabrication factories incorporate several fundamentally different process technologies, and each process technology is capable of being used to make a variety of different devices or products. However, each of these process technologies may use some pieces of fairly expensive equipment also used in other process technologies, in addition to using some process-specific equipment. Hence, the logistics of scheduling is complicated by the need to integrate *all* the processes technologies being simultaneously run. There are few totally isolated processing lines where only one process technology is supported; the norm is a highly complex integration of many interdependent processes.

In addition, each process technology is made up of literally hundreds of individual steps. Some steps require large batches of wafers to be run together for long periods of time (e.g., diffusion furnaces) and some operate on an individual wafer and take but a few minutes (e.g., a wafer stepper). As a result of the fact that some equipment is designed to handle one (or a few) wafers, while other equipment is designed to handle many hundreds of wafers, the balancing of equipment to avoid either under-utilization or potential bottle-necks, is difficult.

Finally, each process technology is itself fairly complicated. All the processes involve lithography; indeed the entire process can be considered to be the integration of a large number (15-20) of lithography steps, where each step uses the same set of equipment as all the other steps, but a different mask or reticle set. Hence, the production line is characterized by a large number of inherently unbalanced steps, using several hundred pieces of equipment that each have different load sizes, and a process recipe that includes as many as 20 loops back through the same processing equipment.

In general, there is an inherent logistical complexity to the scheduling process, due to the number of alternative ways of scheduling lots and equipment to meet business demands, that leads to difficulty in generating "good enough" schedules.

2. Execution complexity

The second issue is quite separate from the first issue, but is adversely impacted by the complexity of the logistics process.

The newest processes are often very difficult to maintain in strict control, since one is working near the theoretical or practical limits of technology. Fifty Angstrom gate oxides, half-micron critical dimensions, very low dopant levels, all represent difficult new processes where both the metrology and control procedures are not well understood. The trend toward smaller feature size continues; it is not likely that the processes used to make semiconductor devices will get any simpler. Unfortunately, much of the processing equipment is relatively unreliable (requiring frequent resetting of process parameters, emergency shutdown, etc.) especially when working near the limits of technology. Hence, the processes are inherently difficult to maintain in control.

As the processes become more complex, and as more steps are added to the process, execution of the schedule over an extended time period, for example, a shift, becomes difficult. Even in simple wafer fabrication factories, there is a lot of process complexity, and in more demanding factories (many processes, many products), the complexity increases dramatically.

Finally, it is often difficult to predict exactly when a piece of equipment will go down, or how long it will take to bring it back up. This problem has two opposite effects; the frequency of going down will decrease, but the uncertainty of the effects of going down will increase.

Hence, the likelihood of the schedule being interrupted by Murphy's Law increases, and scheduling becomes correspondingly more difficult.

There is a process technology influence on scheduling, represented by the fact that much of the technology being used is "state-of-the-art", and every process is inherently "complex", that leads to difficulty in executing schedules.

3. Scheduling strategies

As a result of these two issues, it appears that there is no one scheduling strategy that applies to all situations encountered during the course of a typical day. The main characteristics of many semiconductor manufacturing facilities is change; in process sequences, in product mix, in customer demands, in equipment, etc. In order to deal with all this change, a variety of scheduling strategies need be formulated, where the appropriate strategy can be evoked to meet the particular set of circumstances of the moment. This type of complexity is difficult to deal with using conventional scheduling tools, since these basically are limited in their flexibility.

The nature of the semiconductor business dictates that the "how" of scheduling be adapted to the nature of the events encountered during the establishment of the schedule; scheduling strategies need to be flexible to meet the demands of process and logistic complexity.

Scheduling Hierarchy

Overlaying this scheduling logistics and complexity problem is a scheduling hierarchy issue. Basically, humans have dealt with the complex scheduling problem by partitioning it into manageable chunks. There are several levels of scheduling that can be identified, ranging from the global scheduling needs of the company, down to the dispatch rules given to move a specific lot from one piece of equipment to another piece of equipment at a specific time. We have identified four levels of scheduling, all of which need to be managed:

1. *Factory loading;* the setting of all factory goals based on customer demand and factory capacity and capability. Factory loading is the highest level of planned over a several month time horizon. Plant capacity is determined from analysis of the factory model, the next level down in the scheduling hierarchy.

2. *Factory planning;* the planning of a factory schedule in global terms, depending on the loading passed down from the customer demand. This schedule is the next level down in the hierarchy, and is also run over a many month horizon. During this stage of scheduling, realistic factory behavior needs to be comprehended, via an algorithmic or a simulation model of the factory. The characteristics of the factory are derived from the next level down of the scheduling hierarchy. Furthermore, long-range changes in the factory configuration need to be understood since these can have dramatic effects on factory capacity.

3. *Shift scheduling;* the specific schedule for several shifts, provided in minute-by-minute detail for each piece of equipment and each lot of material. This schedule cannot be accurately derived from simulation of the factory, since a simulator is based on average equipment behavior, and would generate unrealistic specific factory schedules. The shift scheduler is really the key part of the overall scheduler program, since the average behavior of the factory for use in capacity planning is derived from the actual performance of the factory as a result of following the scheduler commands.

4. *Reactive scheduling;* the detection and correction of schedules that become inoperative for any reason (down equipment, lack of a qualified operator, an emergency change in priorities, etc.). This level of scheduler is absolutely necessary to keep the factory running in the event of some unforeseen circumstance, until the shift scheduler is run again.

These different levels of scheduling have different purposes; the characteristics of the four levels of scheduling are shown in the table below.

Level	Key User	Time Horizon	Time Buckets	Time-to-Schedule	Main output
1	Planners	6 months	days to weeks	hours	Capacity plan
2	Manager	months	days to weeks	hours	Factory plan
3	Supervisor	days	minutes	minutes	Schedule
4	Operator	hours	minutes	seconds	Schedule fix

Table 1

Levels of Factory Scheduling

Use of AI in Scheduling

At the present time, we are developing an Artificially Intelligent Scheduler that operates at the factory and shift levels (Levels 2 and 3, Table 1). A Reactive Schedule (Level 4) is being designed, and a Factory Loading Scheduler (Level 1) is under consideration. While none of these modules is yet being applied to schedule production in a factory, the conceptual and technical issues have been resolved, and testing of the modules is taking place.

In view of the nature of scheduling complexity, the four modules listed in Table 1 should be integrated into a global scheduler. In this situation, the scheduling details are consistent from the dispatch level to the factory loading level, giving confidence that the business needs of the corporation have integrity and an excellent chance of being successfully met.

Scheduler Architecture

The Scheduler architecture is illustrated in Fig. 2. In fact, there are two pieces to the Scheduler; an application part, and a architecture part. The architecture part consists of general interfaces to the factory data collection system, interfaces to users, interfaces to factory knowledge management, and interfaces to many possible applications (scheduling is one application; other possible applications are factory layout, yield or cost models, process expertise, etc.).

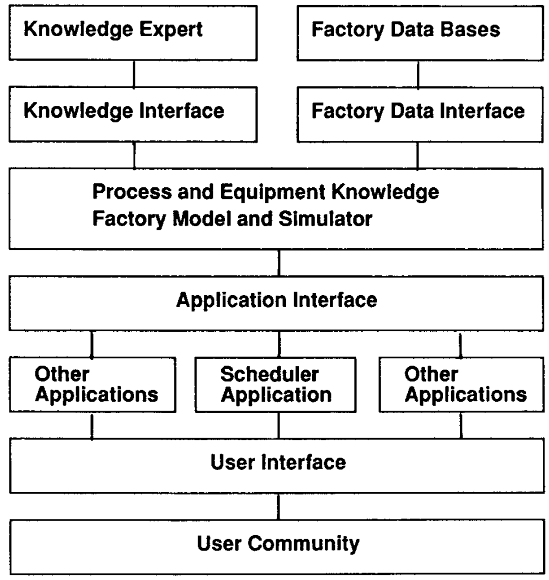


Figure 2

AI Scheduler Architecture

In particular, the Scheduler makes use of a set of interfaces (for users, for domain experts and for data), a model of the factory and a Knowledge Base repository for factory knowledge. Each application engine feeds off the Knowledge Base via a set of application interfaces. The Scheduling Application has all the rules and heuristics required for scheduling the factory at any level. The data interfaces automatically extract data needed to generate a schedule from the on-line shop floor control system and loads it into the factory model. The model digests all this information (about lot priorities, location of all pertinent resources, machine and operator status, etc.) and generates a schedule as a response to a user. The user interface "knows" the characteristics of all the users, and therefore can present each user a schedule tailored to their individual needs. Knowledge about the factory (the characteristics of the equipment, the relationships between various pieces of equipment, the process recipes, etc.) are all managed by a user-friendly Knowledge Base Editor, which makes it possible for non-AI experts to edit the knowledge of the factory.

The knowledge architecture is constructed in such a way that any application will be able to be "plugged in" to the Knowledge Base, in order for there to be a consistent set of data and knowledge that can assist in running a variety of related applications (scheduling, or equipment diagnosis, or yield modeling, or factory layout, etc.).

There are four sub-schedulers required, one for each of the areas in the factory (diffusion, lithography, etch, and thin films). This is necessary for a number of reasons. One reason is simply because the areas are substantially different from the scheduling perspective, with different batching strategies, cycle times, etc. in each area. Another reason is that different problems may arise in the different areas, necessitating different approaches to scheduling during the same time period. The situation in diffusion may indicate a machine-centered approach, while lithography may require a product-centered approach, while etch and thin films may need hybrid machine/job-centered viewpoints. This necessitates an efficient mechanism to interleave the flow between the four areas, and that is this topic which is our current development focus.

Scheduler Operation

The Level 2 AI Shift Scheduler, the "guts" of the AI Scheduler, automatically gathers necessary information from a variety of sources, engages the appropriate scheduling strategy, and then provides a detailed shift schedule for operators (or robots) to apply, until either the schedule is broken, or until the shift is completed and a new schedule is generated. If the schedule is broken, the Reactive Scheduler (Level 4 in Table 1) will be engaged, to provide a short-term schedule "good enough" to be used until a Shift Schedule can be run again.

The data snapshot obtained through the data interface from the factory floor data collection system is used to parameterize the model which the intelligent scheduler will use in its reasoning process. While the model contains a declarative description of the properties of resources (machines, operators, etc.), lots, and process steps, as well as relations between these entities, the data snapshot is necessary to specify precisely the current state of each machine (up, down, if up which lot(s) on which step, etc.) and the identity and condition of each lot (in process, on hold, if in process how much time left on the current step, etc.).

Once the model has been thus instantiated, the intelligent scheduler builds two lists. The first list focuses on resources and has utilization as its metric (for example, bottleneck machines are identified). Computationally, this is not a trivial matter. A brief description of two extreme methodologies demonstrates the difficulty. One way to compute bottlenecks compares the production schedule for the next three months with the machine cycle times corrected for historically averaged down-times. This gives an inherent bottleneck ordering. At the other extreme, a way to compute bottlenecks looks at the work-in-progress levels currently at each machine and the work likely to arrive during the shift. This work is converted to process time and compared to the shift duration. This gives an instantaneous bottleneck ordering. Both orderings are useful in any given context.

The second list which the intelligent scheduler builds focuses on lots and has priority as its metric. While the management level users of the system can indicate which lots they wish to personally prioritize, this still leaves the hundreds if not thousands of remaining lots to be placed in priority order. Again there are a number of possibilities, including lateness from the front of the line, and lateness from the back of the line, to mention two.

These bottleneck and priority lists have two uses in actually building the schedule. The first use is in situation assessment. Inspection of the (instantaneous) bottleneck list can show a fairly smooth ranking with the most loaded resources have acceptable performance, or can show a few machines standing out at the top of the order as being badly overloaded. The latter is an indication of a requirement for machine-centered scheduling, the former is a contra-indication. Likewise with the (lateness from the back of the line) priority list. There can be a few very high priority lots separated at the top of the list, or there can be a uniform ranking with not much difference between the top and the bottom. The former indicates the suitability of job-centered scheduling while the latter indicates unsuitability. Using these kind of heuristics, an approach to the current scheduling problem is selected.

The bottleneck and priority lists are then used to implement the approach. For example, the selection of machine-centered scheduling means that a machine will be selected from the top of the bottleneck list (the most critical resource) and then lots will be selected starting at the bottom of the priority list (the least critical lots) in attempts to fill the machine. Job-centered scheduling begins at the top of the priority list and selects machines starting at the bottom of the bottleneck list. The Gantt chart builder which sits underneath these selection mechanisms works identically in either case, being given lot/machine pairs and trying to place them as early in the schedule as possible. This includes a limited capability to slide already placed lots up and down the time line to support more efficient packing of the resources, again with reference to the priority list.

The information gathered by the Shift Scheduler is used to update the Factory Planning Scheduler, so that the factory loading algorithms represent as close to possible the actual behavior of the factory. These data in turn will be used to load all factories, once models of each factory are completed.

The AI scheduler is capable of dealing with the three issues of complexity discussed previously. Most of the details of all the processes run in the factory can be comprehended by the Scheduler, which includes machine characteristics, the logical sequences of process steps, the people needed to run the equipment, the reticles needed at each step, etc. Indeed, the level of detail can be as great as desired (which, of course, does negatively impact schedule generation time). The Shift Scheduler can keep track of all the details that prevents a human scheduler from performing a comparable job of scheduling, and furthermore provides a bridge over time periods when human communication is weakest (over shift breaks, when shift personnel become ill or are out on vacation, etc.).

Process complexity is also comprehended. For example, the looping of material through a litho area is hard to schedule by a human, but is fairly easily managed by an AI program. Balancing of lines is similarly managed; a set of rules or heuristics tells the Scheduler how to efficiently batch lots, or for how long a piece of equipment should wait for a lot before continuing with the processing step.

The greatest advantage of using an AI-based scheduler is that it has the ability to flexibly schedule fabrication facilities, depending on the nature of the issues to be dealt with on any particular occasion. For example, on one day, the Scheduler may be required to deal with hot lots (because there was an emergency at some customer, or a lot of wafers were below targeted yield, or because there was a new demand for product), while on the next day, the scheduler may be required to deal with machine bottlenecks (perhaps a machine went down for a long repair unexpectedly, or a new machine is being ramped up slowly, or some unforeseen event occurred to clog up a particular process step). In any event, the Scheduler can adapt between the Material-Centered option, or the Machine-Centered option, or, indeed, it may schedule based on some combination of these two extremes. In this sense, the AI Scheduler has the ability to be "clever" in how it schedules; it is not fixed on any one behavior pattern, and can adapt to critical situations accordingly.

The ultimate approach being developed for factory scheduling will use AI technology to actually select an appropriate scheduling strategy from a menu, developed in conjunction with the human schedulers who currently schedule the factory. This will eventually remove the human scheduler from the scheduling loop entirely.

Implementing an AI-Based Scheduler

There are several non-technical problems associated with use of an AI-based scheduling tool in a production environment. These may be grouped into two categories: management education, and management systems.

Management education refers to the problem that many AI solutions have when applied to production rather than to development; there is a healthy skepticism in many conventional computer systems management groups about AI and its failure to produce significant benefits over the last 20 years. Because AI has never had a definition commonly accepted by both the novitiate as well as the sophisticate, there is much contention about what AI really is, what it has really accomplished, and what its major short-comings really are. Consequently, there is a reluctance based primarily on ignorance and fear to using AI technology in a production environment. This reluctance is unfortunately enforced by the failure of the AI hardware and software product vendors to deal with the "problem" of success. What if, somehow, AI development programs were successful; is commercial hardware and software capable of being deployed in a factory? Too often, the answer is no; the hardware does not communicate very well with on-line production hardware, and the software requires a resident LISP programmer in order to preclude expensive shutdowns in case of problems. There has been no real effort on the part of either the computer or the program vendors to make life easier on the production line; most current efforts deal with a development rather than a production environment.

Furthermore, there are often two fundamental misunderstandings of the nature of scheduling that are observed, the further up the management scale (the farther removed from the actual process of scheduling) one progresses. First, it is often the case that senior managers perceive the "absolute" problem of scheduling as trivial, and second, that they perceive the "relative" scheduling of a problem-free fab as "easy", whereas it would be believed that scheduling is important if the factory has lots of issues (many processes or products, many shift breaks, unreliable equipment, market changes, etc.). In fact, scheduling of even a problem-free fab may be easier than scheduling a complex fab, but scheduling is rarely "easy". That is why so much research has been devoted to examining the scheduling problem. This type of issue, the lack of detailed understanding of scheduling complexity issues, is one of the reasons why management is understandably hesitant to accept changes to the current way of doing business.

To all intents and purposes, managing the introduction of AI technology in a manufacturing environment is more a cultural and organizational management issue, and less a technology management issue. Management of change is HARD WORK, regardless of the technology involved.

The only way to deal with the natural resistance to change in every factory, especially in the face of the AI hype seen in the literature and the lack of demonstrable success stories, is to educate management on the reality of applying AI technology in manufacturing. Since most managers mainly fear the unknown, this education (in the form of simple courses in AI concepts, demonstrations of AI capabilities, and support during the transition from conventional solutions to AI-based solutions) is essential.

This leads to the management systems issue. Since one cannot simply implement an AI based solution, relying on vendor software or hardware expertise in case of trouble, one has to set up an internal capability to manage both the hardware and software technology. Such support is common in all conventional computer management systems. Indeed, the support capability has had to evolve from support of one computer system (say a VAX system) and one language (say COBOL) to a heterogeneous environment that may consist of DEC, HP and IBM systems that are expected to communicate with each other using many languages (FORTRAN, COBOL, Pascal, "C", etc.). Adding "AI languages" such as LISP, and AI hardware such as Symbolics or SUN workstations, in reality only adds a small degree of complexity, but in perception, becomes a major obstacle. There are few LISP programmers around that want to provide support (vs working in the development of new AI applications) and the hardware support for sophisticated AI workstations located outside of major metropolitan areas, is weak. Consequently, for the next few years, it is expected that support for both the hardware and software for non-expert system AI applications will have to come from internal sources such as an AI technology support team. Such a concept needs to be discussed with the factory Information Systems personnel, so that they eventually take over these support responsibilities.

Conclusions

The issues found in developing an AI-based scheduler seem to be common to the development of any new computer system, but in addition, there is a "mystique" associated with AI technology that challenges development engineers, excites potential users of AI applications, and terrifies management. It does appear as if AI applications will start to appear in factory environments in greater numbers, and consequently, there is a need to deal with the issues raised.

It is important to recognize that AI technology IS different from conventional computer technology; that programming is different, that documentation is different, that user interfaces are different, that support and training are different. For example, many AI software programs are NEVER finished; there is ALWAYS something that can be added to the program to make it behave more appropriately, according to the programmer. This is to be looked at as an advantage to AI systems, rather than a detriment, since the AI programs are being implemented because of their flexibility and scope. Hence, AI programs do require constant attention and updating; their knowledge bases must be kept current, or else they grow obsolete and the program usefulness diminishes. Lots of the bad publicity for AI came about specifically because of this problem; the users thought they wanted a rigid turn-key system, and were ill prepared to support the adaptive and highly interactive system they received.

AI technology needs internal champions, lots of salesmanship, and lots of nurturing, before it can be perceived as beneficial in any factory. As much attention needs to be paid to the gaining of confidence of the factory personnel, as does the technology. While it is true that poor technology never is successful, it is equally true that superior technology is all too often unsuccessful because it was not managed properly in the eyes of the ultimate user. Establishing the right level of expectations, understanding the customer needs, making sure the customer believes that you understand their needs, and then supplying technology that satisfies the customer expectations, is all necessary to successfully deploy artificially intelligent applications in a production environment. The investment in doing all this is heavy, but the benefits, if done properly, far outweigh the investment costs.