

HIGH-FIDELITY RAPID PROTOTYPING OF THE REAL-TIME FAB OPERATIONS THROUGH DISCRETE EVENT SYSTEM MODELING

Jonghun Park, Spyros Reveliotis, Douglas Bodner, Chen Zhou, Jenn-Fong Wu and Leon McGinnis
School of Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332
e-mail: spyros@isye.gatech.edu

KEYWORDS

High-fidelity fab modeling, flexible automation, real-time shop-floor control, Web-based simulation

ABSTRACT

Traditionally, the semiconductor manufacturing industry has been driven by continuous technological advancement of the underlying production processes. Yet, as the industry matures, mere technology development is no longer sufficient. The effective deployment and exploitation of the system production capacity and operational capability become critical for competitive success. Hence, currently, there is an increasing interest towards the development of a control paradigm / framework that will allow the effective and efficient (re-)deployment and operation of contemporary fabs. The research program presented in this paper seeks to define a detailed modeling and control framework for the real-time fab operations, by exploiting and integrating emerging results in Discrete Event Systems theory. Furthermore, in addition to the development of the formal specification, the presented program will also implement the proposed fab modeling and control framework in a Web-based simulation platform, that will function as a fab (re-)configuration and control synthesis tool for the fab control engineer, and as an educational tool for manufacturing system modeling and control.

INTRODUCTION

Modern high-tech industry is characterized by a number of trends that affect the way in which manufacturers produce finished products. In today's environment, manufacturers achieve competitive advantage by offering a variety of product types, a high level of product quality, and short order lead times (i.e., the time from which an order is placed until the product is received). In addition, there is a constant pressure for innovative product design and customized products, which results in compressed product lifecycles and the need for facility (re-)configuration

to support the next generation of product. These trends contrast significantly with the traditional paradigm of mass production, which places its emphasis on efficient production of high volumes of a standardized and relatively stable product type.

The semiconductor industry is a prime example of such trends. Manufacturers produce a variety of semiconductor-based products or integrated circuits (e.g., memory chips or processors). At the same time, product quality is a critical consideration. Semiconductor material is sensitive to the slightest contamination; hence, manufacturers employ extensive automation to ensure contamination-free, or clean room, environments. To meet customer lead time demands, manufacturers focus on reducing cycle time – i.e., the time that it takes to produce a finished product once the raw material has been released for production – and cycle time variability. From a planning perspective, reduced customer lead times and increased system responsiveness require the continuous (re-) alignment of the planned production activity to externally imposed demand. Finally, manufacturers are faced with the constant challenge of having to adapt and produce new types of integrated circuits, given the rapid pace of technological advancement in product design (NEMI 1998).

In the past, the semiconductor manufacturing industry has tried to address these market challenges primarily through technological innovation. For example, most major improvements in system performance have occurred from: (i) the stabilization of the production processes, which increases production yields, (ii) the development of integrated, highly controlled production environments (Bader et al. 1990), such as cluster tools (Singer 1995), that reduce the time wasted in unproductive wafer transfers, and the risks arising from their exposure to environmental contaminants, and (iii) the continuous reduction in feature size, which increases the number of integrated circuits produced per wafer, and eventually, the system throughput. At the same time, there has been considerable effort to promote the hardware-based integration of the shop-floor activity. More specifi-

cally, the industry has actively pursued: (i) the deployment of powerful manufacturing execution systems (MES), that enable the real-time monitoring of the fab operations, and (ii) the development and adoption of standard operational models and protocols, like GEM and SECS/SECS II, that facilitate the easy and seamless interface between supervisory controllers and tools (SEMI 1999; Du and Zhou 1999).

However, currently the industry is lacking a control paradigm that will take advantage of the aforementioned real-time shop-floor monitoring and command capability, and will provide an integrated framework for the efficient and holistic management of the flow of product, material, information, people, and other resources. Most prior attempts to address the real-time production control in the fab operational context have focused exclusively on the "scheduling" problem, i.e., the development of part-flow control policies, that when imposed on the system operation, lead to high production throughput through effective utilization of the system bottleneck resources. Typically, these policies are structured as a set of (i) job release (wafer-start) and (ii) job dispatching heuristically developed rules for controlling the part flow among the different bays and/or bay processes (Kumar 1994a; McCutchen and Lee 1995). The efficiency of these rules is established / claimed on the basis of (i) accumulated past experience, (ii) (simulated) experimentation in the particular fab context (Patel 1999; Zoghby 1999), and, in certain cases, (iii) their proved optimality in the context of some simplifying analytical characterizations of the fab operations (Kumar 1994b). It is also noticed that some more involved scheduling schemes, attempting a more global view of the system operations, and a more systematic performance optimization, have been proposed in the literature (Connors et al. 1994; Sharifnia 1995), but their effectiveness and implementability, especially in an extensively automated production environment, remains to be seen.

Indeed, in order to successfully deploy large-scale flexible automation, especially through the advent of 300mm wafer manufacturing technologies, new methods are needed for systematic characterization, analysis and control of the underlying system behavior, from the perspective of logical correctness and consistency (PRI Automation - personal communication). The resulting problems are collectively characterized as the logical / structural control problem of the fab operations, and they can be perceived as complementary to the more traditional performance-related fab scheduling problem (Bodner and Reveliotis 1999). Moreover, to the extent that structural

control guarantees the correctness of the system behavior, it defines its admissible behavioral space, and therefore, any performance optimizing policy should be aligned with the structural control policy (SCP) requirements. Yet, in the past, logical / behavioral problems arising in the real-time control of the fab environments have been largely ignored during the system design and development, and they have been resolved mainly through external (human) intervention. More recently, there has been some effort to address these problems during the controller design phase, but the currently implemented policies seem to be completely customized to the particular fab operation, developed in an ad-hoc fashion, and validated only through custom-supported, simulated experimentation (Patel 1999; Zoghby 1999). Hence, there is no rigorous verification of the correctness of the developed control logic, and more importantly, the resulting controller cannot be easily reconfigured under changing circumstances. The bottom line is a user-unfriendly controller, characterized by high developmental costs and operational inflexibility.

Motivated from the above remarks, the work described in this document seeks to develop a high-fidelity, rapid prototyping framework, appropriate for the systematic modeling, analysis and control of the real-time fab operations. Specifically, our work employs formal models - Colored Petri Nets (CPN) (Jensen 1997) - from the burgeoning field of Discrete Event Systems (DES) theory (Cassandras and Laforune 1999), in a hierarchical architecture, that allows the detailed modeling of the fab activity at the equipment / workstation, bay, and shop-floor level. The developed models allow for the representation of equipment behavioral aspects like batching, setups, loading / unloading procedures, and machine failures. Special attention is given to modeling previously ignored aspects, like (i) the internal behavior of the inter and intra-bay material handling systems (MHS), and (ii) effects resulting from the behavior of the supporting staff, like the maintenance personnel, and the equipment loading / unloading operators. Kempf's mini-fab specification (Kempf 1994) is currently used for defining the detailed model specifications, and for assessing the representational adequacy of the proposed framework.

The formal characterization of the (uncontrolled) fab behavior subsequently allows: (i) its formal analysis through emerging DES theory, and (ii) the development of control policies that are able to establish the logical consistency / correctness and efficiency of the fab operations. Specifically, this part of the work leverages the aforementioned fab models in order to develop a control architecture for the real-

time fab operations, addressing logical concerns - like deadlock-free buffer space allocation and/or collision-free vehicle traveling in the fab MHS's - as well as efficiency considerations - like production throughput maximization and reduced cycle times. To this end, we employ results from the recently emerging area of structural control of resource allocation systems (RAS) (Reveliotis 1996), and scheduling theory (Uzsoy et al. 1994; Kumar 1994a). Our ultimate objective is the development of a Web-based computational platform that will facilitate the effective and efficient deployment of fab operations and their underlying control logic, through the integration of structured fab modeling, control synthesis, validation of the controlled system behavior, and evaluation of its performance, by means of available analytical techniques and / or simulated experimentation. A pictorial representation of the undertaken research program and of the way in which it seeks to integrate and promote existing analytical results, is depicted in Figure 1.

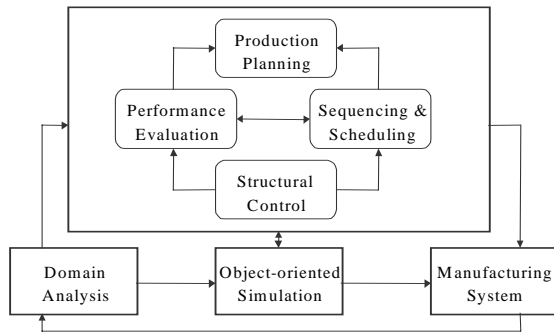


Figure 1: The proposed research program structure.

The rest of the paper is structured as follows: Section 2 delineates the proposed fab modeling framework, discussing the underlying modeling principles, the employed modeling abstractions / entities, and their presumed relationships. It also maps the proposed fab behavioral model(s) to a hierarchical control framework, consisting of a series of control sub-problems that correspond naturally to the basic fab modeling entities. Finally, it indicates how the controller synthesis for the identified control (sub-)problems can be supported through existing literature results, and how these partial solutions can be integrated to a seamless control framework for the entire fab operation. Section 3 discusses the software architecture of the proposed Web-based simulator, and its intended functionality. Section 4 concludes the paper by discussing the current progress of our efforts, and the intended future work.

THE PROPOSED FAB MODELING FRAMEWORK AND THE UNDERLYING CONTROL PROBLEM

The purpose of our work is to support easy development / reconfiguration of real-time controllers for the flexibly automated fab, and assessment of the controlled system robustness and performance, by focusing on the logistics of material flow and process deployment. Some specific principles that have driven our work are: (i) the developed modeling methodology should be as simple as possible, i.e., it should require the fewest number of entity types and relationship types; (ii) the developed modeling methodology should permit a high degree of model configuration and customization, to accommodate a wide variety of applications; (iii) the resulting models should map naturally onto real fab environments, i.e., the abstraction gap between reality and the model should be as small as possible.

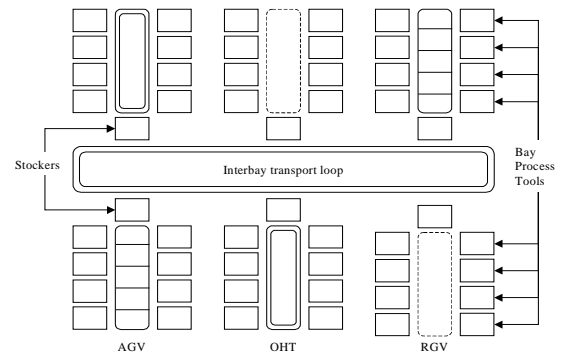


Figure 2: The typical layout for the emerging 300mm semiconductor fab.

Manufacturing systems are often viewed as hierarchical organizations. Although this is not the only way to view such systems, currently it is the most conventional, and it applies naturally to the currently adopted fab layouts. As a result, it constitutes the basis of many existing control implementations, and it is the view adopted in this work. As a case in point, consider the fab layout illustrated in Figure 2. This layout is a typical layout contemplated for the emerging 300mm fab (Cunningham et al. 1999), and it consists of a number of bays, stemming from a central aisle, which gives it the characterization of a spine configuration. The central aisle contains the interbay MHS, typically an overhead transport system, transferring cassettes of wafers among the fab bays. The bays themselves are interfaced to the interbay MHS through the bay stocker, i.e., an automated storage / retrieval system (AS/RS), which accommodates wafer cassettes waiting for processing in the

bay area, or cassettes that have completed processing in the current bay, and request transferring to some other bay. Within the bay, process tools are also organized around another automated material handling system, typically an Automated Guided Vehicle (AGV), Rail Guided Vehicle (RGV) or Overhead Hoist Transport (OHT) system, facilitating intrabay part transfers. Depending on the specific layout adopted, process tools in a certain bay may be aggregated on the basis of a common functionality (farm layout), or the support of a certain processing sequence (modified hybrid layout). Quite often, a third layout, known as hybrid layout, is obtained from the farm layout, by adding into each bay area the metrology tools required for evaluating the outcome of the bay processes.

For the modeling purposes pursued in this work, the fab layout illustrated in Figure 2 is viewed as having the hierarchical structure displayed in Figure 3. Specifically, our modeling methodology represents each element in the hierarchy as a "control domain", with a particular structure. Furthermore, adopting the modeling perspectives of the prior OOSIM effort towards modeling discrete manufacturing systems (Narayanan et al. 1994; Bodner 1996), the modeling entities characterizing the behavior of a certain control domain can be discriminated on the basis of (i) processing vs. logistics, and (ii) plant vs. control.

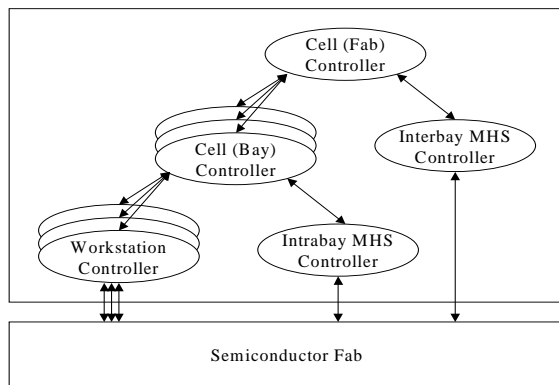


Figure 3 The proposed hierarchical modeling and control architecture

The fundamental abstractions supporting the "plant" view of the fab operations are material and location, defined as follows: (i) Material is a discrete job entity that is transported, stored and processed in the considered controller domain(s). A material unit has a unique identity, as well as handling and processing requirements, which constitute the process plan / recipe associated with it. Depending on the specific fab operational context, a material unit can

correspond to a single wafer, or a cassette of wafers collectively processed through the system as a wafer lot. (ii) Location is a specific identifiable physical space where material may reside while it is being processed, transported or stored. A location has capacity (in terms of wafers or wafer cassettes), and a capability (i.e., a set of functions that it can perform, each of which may have associated with it some processing time requirements). A location is also characterized by a state, determined by: (a) its current content (e.g., a job with a specific ID), and (b) its status (e.g., idle, busy, failed, etc.). Furthermore, each location has a "behavior" that can be characterized analytically. Hence, a process tool can be abstracted to a location that transforms wafers (material) from one state (prior to an operation) to another state (after an operation). An inspection station transforms material logically, by adding information about the condition of the material. Storage systems consist of locations that transform material in time. A robot or an AGV is a location that may contain a number of units of material, and transforms them in terms of both spatial and temporal state.

In our approach to fab modeling, the control system is represented by a set of entities that are distinct from their associated plant entities. These entities respond to changes in the state of material and locations of their associated plants, by initiating certain behaviors at the plant locations. The control system, then, is naturally described in terms of collections of locations that are managed by a specific controller. We will refer to such collections as cells. In general, a cell will contain two types of locations, those that transform the physical or information properties of the material, and those that transport or store the material. This distinction is captured in the processing vs. logistics view, in which the cell consists of two types of subsystems: (i) The cell production subsystem, i.e., the set of locations performing physical or logical transformations of material. These locations have an associated operation-location map, which specifies the set of production locations that can perform a particular operation, or step in a process plan. (ii) The cell MHS, i.e., the set of locations associated with storage of material and movement operations to transfer material between production locations, storage locations, and cell input/output locations. A domain map specifies the set of routes associated with each material handling device or location.

In the context of the fab operations illustrated in Figure 2, and depending on the level of abstraction in the fab modeling hierarchy, a cell can model the operation of a cluster tool, a bay area, or the entire fab shop-floor. Its physical domain is defined by the

AS/RS's (stockers) interfacing the cell production units to the inter-cell MHS, and the AS/RS owned by the cell itself. The cell supervisor coordinates the part flow within the cell production subsystem, i.e., the induction of new jobs into the cell, and their dispatching to the required cell production units. On the other hand, the internal operation of each of the cell production units and its MHS is controlled by dedicated workstation (or cell, depending on the complexity of internal material flow of the production unit) and MHS supervisors. These supervisors are in subordinate relationship to the parent cell controller, in the sense that (i) their processed tasks / jobs are defined by commands generated by the parent cell supervisor, and (ii) they provide feedback information to it regarding the processing status of their task assignments. In turn, these lower level controllers supervise the operation of the hardware devices and/or logical system components in their domain, by exchanging a series of communication signals with the controller in these units. Hence, given the regularity characterizing the proposed command hierarchy, it can be concluded that the entire control architecture is completely defined by specifying: (i) the generic internal control logic for (a) the system workstation controllers, that constitute the leaf entities of the proposed command hierarchy, (b) the intra-cell MHS controller, and (c) the cell production system controller; (ii) the communication messages exchanged between the cell controller, and its subordinate production unit and MHS controllers.

The development of a generic control logic for the three controller classes identified above, can be further systematized by noticing that the control problem underlying each class can be (abstractly) described as the problem of supporting the concurrent execution of a sequence of externally specified tasks, through logically correct and efficient allocation of the limited resources of the corresponding controller domain. Hence, for the purposes of the further analysis and controller design, it is pertinent to abstract each of the identified controller domains to a resource allocation system (RAS) (Reveliotis 1996). This abstraction facilitates the application in the underlying controller design of more general methodological approaches proposed for DES analysis and control (Cassandras and Lafortune 1999). Specifically, according to the RAS theory developed in (Reveliotis 1996), a RAS model can be further represented as a controlled finite state automaton (FSA) (Cassandras and Lafortune 1996). In this FSA model, the RAS state is essentially the composite of the states of the locations contained in the RAS domain, while state transitions correspond to job initiations, terminations, and advancements between their proc-

essing stages, commanded by the domain controller to its subordinate controllers, which execute and confirm the corresponding task(s). From a physical standpoint, the material transfer among the different control sub-domains of a given cell supervisor- i.e., the units defining the cell production subsystem and the MHS – is facilitated through a set of locations modeling the buffering capacity of the access points of these units. Since these locations define the interface points between the considered control domain and its subordinate control domains, they are characterized as shared locations. Then, from an information-flow / logical perspective, the problem of establishing effective communication / coordination among the different system controllers can be posed as the requirement for effective management of the shared locations interfacing the controller domains. At the same time, it should be realized that for the FSA modeling the behavior of the considered controller domain, the sequence of the communication exchanges among the domain RAS controller and its subordinate controllers define the set of events that can be experienced by the RAS controller, and therefore, they determine its perception capabilities regarding its underlying operational domain, its discriminatory power, and eventually, the set of its potential behaviors. One of the major efforts undertaken by our work is the definition of the messages to be exchanged between the various system controllers, and of a supporting communication protocol that will ensure logically correct and robust behavior for the entire flexibly automated fab shop-floor.

The FSA representation of the operation of the controller domains defined in Figure 3 is also instrumental for structuring the underlying decision making / control process. Figure 4 illustrates the proposed generic control scheme, in which the controller decision making process is triggered by the occurrence of (communication) events that either (i) inform the controller about changes taking place in its environment, or (ii) command actions imposed by the unit controller on its domain. All these events are reflected as changes in the perceived RAS state. Furthermore, the system controller reacts to a certain event by: (i) evaluating the set of admissible actions among the set of feasible actions associated with the perceived RAS state resulting from the communicated event, and (ii) selecting a specific admissible action for immediate execution by the control domain. This selection is reflected in the data structure representing the system state, and the entire cycle repeats itself.

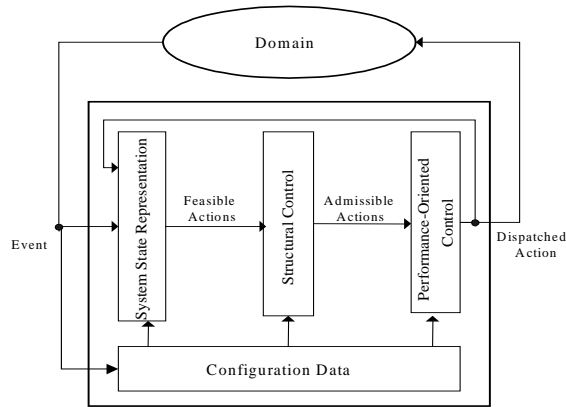


Figure 4: A generic real-time event-driven RAS control scheme

The part of the RAS control logic that determines the admissibility of any feasible control action in a given RAS state is characterized as the structural control policy (SCP). Currently, the most conspicuous problem addressed by structural control in contemporary flexibly automated production environments is the resolution of the potential system deadlocks arising from the competition of a number of concurrently executing jobs for the finite set of the system resources. Computationally efficient policies that have been rigorously shown to establish deadlock-free system operation under a series of RAS behavioral contexts, can be found in (Reveliotis and Ferreira 1996; Reveliotis et al. 1997; Lawley et al. 1998a, Lawley et al. 1998b; Reveliotis 2000a). More recently, the work presented in (Park et al. 1999) has extended and integrated these results so that they apply in the distributed real-time control architecture presented here. Specifically, Park et al. (1999) use the formalism of communicating colored Petri nets (CPN) in order to characterize the underlying domain / RAS behaviors, and the imposed structural control logic. Currently, we work towards extending the basic RAS models defined in (Park et al. 1999), in order to accommodate additional operational features pertaining to the fab operations, including batch-based processing at certain stations, sequence dependent setup, loading / unloading procedures requiring also the presence of a human operator that serves a number of stations, and machine failures.

The part of the RAS control logic driving the selection of a specific action among the set of admissible actions to be commanded for immediate execution, defines the performance-oriented scheduling policy implemented by the controller. Most dispatching / scheduling policies existing in the literature (e.g., the dispatching rules of (McCutchen and Lee 1995), the distributed scheduling policies of

(Kumar 1994b), and the more computationally involved tracking policies of (Sharifnia 1995)) can be easily implemented in the context of the control logic depicted in Figure 4. However, some recently obtained results (Reveliotis 2000b) indicate that the effectiveness and efficiency of the aforementioned scheduling policies in the contemporary flexibly automated fab is an open research issue. We believe that the representation of the fab operations developed in this paper, and the associated control architecture, provide the appropriate context for addressing the underlying performance evaluation and control problem. Yet, the resulting control problem is essentially equivalent to the notorious “job-shop scheduling” problem (Sule 1997), reformulated in the operational context of the proposed framework, and therefore, its consideration is deferred to future work. In the meantime, the high-fidelity Web-based simulator of the real-time fab operations, outlined in the next section, should function as a compromise solution towards the systematic fab performance evaluation, and the synthesis of efficient, heuristic-based fab scheduling policies.

A WEB-BASED SIMULATION PLATFORM FOR THE PROPOSED FAB MODELING AND CONTROL FRAMEWORK

The application of discrete event simulation for (i) the empirical validation of the correctness of synthesized real-time control policies, and (ii) the evaluation of the resulting system performance, is already an established practice in the semiconductor manufacturing industry. Currently, however, the development of the fab operational model and the synthesis of the control policies is more of an art rather than a structured engineering process, and the deployment and support of the simulation capability almost exclusively requires the employment of externally hired simulation experts. Furthermore, given the capabilities of the existing commercial simulation platforms, in many cases, the underlying fab control logic is confounded with the representation of the physical and logistical operations, and therefore, it is only partially or implicitly stated in the resulting software model. As a result, the development of high-fidelity simulation models for fab operations is expensive, time consuming and quite complicated. Moreover, the end product is rather esoteric, and therefore, not easily manageable by the shop-floor control engineers (NEMI 1998).

Motivated by the above remarks, our proposition is the development of a Web-based simulator that will provide a structured and integrated environment

for the modeling, analysis and control of the (real-time) fab operations, based on the fab modeling, behavioral analysis, and control synthesis framework presented in Section 2. More specifically, the simulator user will interact with the software through a client web-based application, and she will be able to (i) configure the simulated fab model in terms of a number of parameters / features, including the fab, bay and the inter/intra-bay MHS layouts, and subsequently (ii) define the imposed structural and performance control policies. Hence, following the modeling principles outlined in Section 2, the proposed simulation environment will explicitly distinguish between the plant operational model, and the applied control policies. Furthermore, following the generic scheme of Figure 4, the control logic itself will be differentiated into logical / structural and performance-related policies, with the former addressing admissibility decisions relating to the establishment of deadlock-free material flow, conflict-free vehicle routing, consistent tool / reticle management and acceptable batching patterns, and the latter addressing scheduling decisions, like job induction and dispatching to the different system processing units, efficient vehicle allocation to job transfer requests and routing, and efficient tool / reticle allocation to the requesting processes. Finally, the simulator will provide for the specification of maintenance policies, and the handling of exceptional situations like equipment failures and the accommodation of expedient jobs.

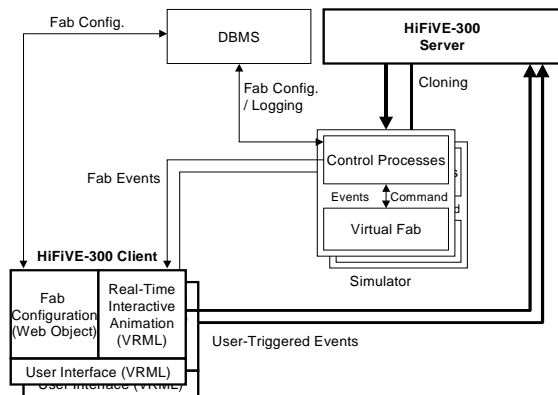


Figure 5: The HiFiVE-300 architecture

Once the user has configured the fab model and the underlying operational logic, the data will be stored in a database system, remotely located on the simulation server. Upon the user / client's command, the server will initiate the simulation of the real-time fab operations, while the simulation output (events) will be communicated to the client through CORBA middleware (Orfali and Harkey 1998), and it will drive a

client-based VRML animation (cf. <http://www.vrml.org>). The simulation itself will be distributed, consisting of a number of communicating concurrent processes, corresponding to the fab operational units / control domains and their associated controllers. In this way, it would be conceptually possible to (eventually) substitute plant modeling units with their corresponding actual processes, converting the simulator to a (real-time) controller. Under more conventional usage, the user will be able to obtain a number of performance indices, including flow time, resource utilization and yields. Furthermore, the user will have the capability to interact with the simulation in real-time, by triggering a number of events like machine failures / repairs, and the arrival of expedient jobs, and thus, she will be able to test the (controlled) system behavior under a number of dynamically generated scenarios. A schematic representation of the proposed software architecture – to be called the High Fidelity Virtual Environment –300 (HiFiVE-300) architecture – is given in Figure 5. The formal and structured modeling approach underlying the software development, combined with the application of prevailing object-oriented technologies (Wirfs-Brock et al. 1990) in the software implementation, will also facilitate the maintenance of the proposed simulation platform in terms of modification and/or inclusion of new basic component behaviors and control policies.

DISCUSSION

The research program outlined above is a major effort undertaken by the Virtual Factory Lab (VFL), in the School of Industrial & System Engineering, at Georgia Institute of Technology. Currently we are in the stage of detailing and validating the formal models – Colored Petri Nets (Jensen 1997) – representing the fab operation at the workstation, bay and factory level, in the proposed hierarchical framework. Kempf's mini-fab specification (Kempf 1994) is our current reference point for the real-time fab operation. We would like, however, to elicit the collaboration of any interested industrial parties towards the more extensive revision and/or enhancement of the modeled features and behaviors. Subsequent steps will involve the software implementation of the parameterized models, their integration in the architecture of Figure 5, and the development of the user interfaces that will allow the effective and efficient utilization of the developed simulation capability by the fab control engineers and/or the interested educational audiences.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the W. M. Keck Foundation and of the NSF - Grant no. DUE-9950301 - for the presented research effort.

REFERENCES

- Bader, M. E., Hall, R. P. and Strasser, G. 1990. "Integrated Processing Equipment", *Solid State Tech.* 33, no. 5: 149-154.
- Bodner, D. A. and Reveliotis, S. A. 1999. "Flexible Semiconductor Manufacturing". In *Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster, ed. John Wiley & Sons, Vol. 7: 596-607.
- Bodner, D. A. 1996. "Real-time control approaches to deadlock management in automated manufacturing systems". *PhD Thesis*, School of Industrial & Systems Eng., Georgia Tech, Atlanta, GA.
- Cassandras, C. and Lafortune, S. 1999. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers.
- Connors, D., Feigin, G. and Yao, D. 1994. "Scheduling semiconductor lines using a fluid network model", *IEEE Trans. on R&A* 10: 88-98.
- Cunningham, C., Wright, R., Benhayoune, K., Campbell, E., Swaminathan, V. and White, R. 1999. "300mm factory layout and automated material handling system analysis". In *Proceedings of the AutoSimulations Symposium '99* (Salt Lake City, UT). AutoSimulations.
- Du, H. and Zhou, C. 1999. "Message-oriented decomposition for supervisory control in manufacturing system", *Robotics and Computer Integrated Manufacturing* 15, no. 6: 441-452.
- Jensen, K. 1997. *Coloured Petri Nets: Volume 1* (2nd ed.). Springer, NY, NY.
- Kempf, K. 1994. "Intel five-machine six-step mini-fab description". Technical Report: http://www.eas.asu.edu/~aar/research/intel/papers/fab_spec.html, Intel.
- Kumar, P. R. 1994a. "Scheduling semiconductor manufacturing plants", *IEEE Control Systems Magazine* 14, no. 6: 33-40.
- Kumar, P. R. 1994b. "Scheduling manufacturing systems of re-entrant lines". In *Stochastic Modeling and Analysis of Manufacturing Systems*, D. D. Yao, ed. Springer-Verlag.
- Lawley, M. A., Reveliotis, S. A. and Ferreira, P. M. 1998a. "A correct and scalable deadlock avoidance policy for flexible manufacturing systems", *IEEE Trans. on R&A* 14: 796-809.
- Lawley, M. A., Reveliotis, S. A. and Ferreira, P. M. 1998b. "The application and evaluation of Banker's algorithm for deadlock-free buffer space allocation in flexible manufacturing systems", *Intl. Jnl of FMS* 10: 73-100.
- McCutchen, T. and Lee, C-Y. 1995. "An analysis of dispatching rules in a semiconductor wafer fabrication environment", *Jnl of Electronics Manufacturing* 5: 165-174.
- Narayanan, S., Bodner, D. A., Sreekanth, U., Govindaraj, T., McGinnis, L. and Mitchell, C. 1994. "Modeling control decisions in manufacturing systems simulation using objects". In *Proceedings of the 1994 IEEE Intl. Conf. on Systems, Man & Cybernetics*, IEEE, 1392-1397.
- National Electronics Manufacturing Initiative (NEMI), Inc.: "*National Electronics Manufacturing Technology Roadmaps, 1998.*"
- Orfali, R. and Harkey, D. 1998. *Client / Server Programming with Java™ and CORBA* (2nd ed.) John Wiley & Sons, NY, NY.
- Park, J., Reveliotis, S. A., Bodner, D. A. and McGinnis, L. 1999. "A distributed event-driven control architecture for flexibly automated manufacturing systems". *Tech. Report* (submitted to IJCIM). School of Industrial & Systems Eng., Georgia Tech, Atlanta, GA.
- Patel, N. 1999. "Using simulation to evaluate complex control logic for lot delivery". In *Proceedings of the AutoSimulations Symposium '99* (Salt Lake City, UT). AutoSimulations.
- Reveliotis, S. A. 1996. "Structural analysis and control of flexible manufacturing systems with a performance perspective". *Ph.D. Thesis*. Dept. of Mechanical and Industrial Eng., University of Illinois at Urbana-Champaign, Urbana, IL.

Reveliotis, S. A. and Ferreira, P. M. 1996. "Deadlock avoidance policies for automated manufacturing cells", *IEEE Trans. on R&A* 12: 845-857.

Reveliotis, S. A., Lawley, M. A. and Ferreira, P. M. 1997. "Polynomial complexity deadlock avoidance policies for sequential resource allocation systems", *IEEE Trans. on Automatic Control* 42: 1344-1357.

Reveliotis, S. A. 2000a. "Conflict resolution in AGV systems", *IIE Trans.* (to appear).

Reveliotis, S. A. 2000b. "The destabilizing effect of blocking due to finite buffering capacity in multi-class queueing networks", *IEEE Trans. on Automatic Control* (to appear).

Semiconductor Equipment and Materials International (SEMI) 1999. SEMI E30-95: *Generic Model for Communications and Control of SEMI Equipment (GEM)*.

Sharifnia, A. 1995. "Stability and performance of a simple distributed tracking policy for production control of manufacturing systems", *IEEE Trans. on Automatic Control* 40: 1109-1113.

Singer, P. 1995. "The driving forces in cluster tool development", *Semiconductor Int.* 18, no. 1: 113-118.

Sule, D. R. 1997. *Industrial Scheduling*. PWS Publ. Co., Boston MA.

Uzsoy, R., Lee, C-Y and Martin-Vega, L. A. 1994. "A review of production planning and scheduling models in the semiconductor industry; part II: shop-floor control", *IIE Trans.* 26: 44-55.

Wirfs-Brock, R., Wilkerson, B. and Wiener, L. 1990. *Designing Object-Oriented Software*. Prentice Hall, Englewood Cliffs, NJ.

Zoghby, J. 1999. "Improving planning decisions through AutoSched AP". In *Proceedings of the Auto-Simulations Symposium '99* (Salt Lake City, UT). AutoSimulations.