Flexible Experimentation and Analysis for Hybrid DEVS and MPC Models

Dongping Huang, Hessam S. Sarjoughain, Gary W. Godding

Arizona Center for Integrative Modeling & Simulation
Computer Science & Engineering Department
Arizona State University, Tempe, Arizona

Daniel E. Rivera

Control Systems Engineering Laboratory
Chemical Engineering Department
Arizona State University, Tempe, Arizona

Karl G. Kempf

Decision Technologies
Intel Corporation, Chandler, Arizona

ABSTRACT

Discrete-event simulation and control-theoretic approaches lend themselves to studying semiconductor manufacturing supply-chain systems. In this work, we detail a modeling approach for semiconductor manufacturing supply-chain systems in a hybrid DEVS/MPC testbed that supports experimentations for DEVS and MPC models using $KIB_{DEVS/MPC}$. This testbed supports detailed analysis and design of interactions between discrete processes and tactical controller. A set of experiments have been devised to illustrate the role of modeling interactions between Discrete Event System Specification and Model Predictive Control models. The testbed offers novel features to methodically identify and analyze complex model interactions and thus support alternative designs based on tradeoffs between model resolutions and execution times.

1 INTRODUCTION

A variety of simulation and optimization approaches have been proposed for semiconductor manufacturing supply-chain systems for many years (e.g., Shapiro 2001 and Kempf 2004). Each approach aims at tackling specific aspects of semiconductor enterprise systems. For example, Discrete-Event Simulation (DES) is beneficial for capturing realistic dynamic behaviors. Linear Optimization (LP) and Model Predictive Control (MPC), on the other hand, are suitable for modeling strategic planning and tactical control. More recently, due to the increasing complexity in semiconductor enterprise-level operations (e.g., on-demand manufacturing, short prod-

uct shelf-life, distributed operations, and lower cost), synthesis of the complementary modeling approaches has been attracting researchers and practitioners (Gjerdrum et al. 2001, Godding et al. 2004, Venkateswaran and Jones 2004, Sarjoughian et al. 2005, Xu and Sen 2005, Zeigler 2006).

Formalizing the composition is desirable to achieve systematic synthesis of disparate models. Advances in simulation interoperability and software engineering—e.g., HLA (HLA 2000), agent-based modeling (Gjerdrum et al. 2001), and software service-oriented architecture (Tsai et al. 2006)—support integrating different simulation models as independent components and enable interactions between them. These approaches and their supporting technologies employ low-level programming languages or high-level interoperability concepts to allow simulators to interact. Such approaches, however, are not based on model composability. Moreover, lack of model composability concepts and methods adversely affects capturing domain-specific knowledge.

A modeling composition approach referred to as Knowledge Interchange Broker (KIB)—was developed to formalize integration of disparate models at the level of modeling formalisms (Sarjoughian and Plummer 2002). The conceptual basis of KIB is that disparity between different syntax and semantics should be accounted with a distinct model and thus enabling independent data and control interactions required for simulation models to interact. In particular, rather than relying on middleware concepts and techniques, interaction among models is specified as a pair: (i) model composability at the level of modeling formalisms and (ii) simulation interoperability at the level of simulation execution. The separation of

the disparate models and their composition enables two key activities of simulation model development —i.e., model validation and simulation verification.

Hybrid discrete-event simulation with model predictive control (Sarjoughian et al. 2005) and linear optimization (Godding et al. 2004) have been developed using the KIB approach. The realization of these approaches use DEVSJAVA (ACIMS 2001) as the discrete-event simulation tool, SIMULINK/MATLAB as the MPC tool (Mathworks 2005), and OPL Studio as the optimization tool (ILOG 2005).

1.1 Related Work

As noted in the previous section, multi-modeling approaches are becoming increasingly popular for simulating integrated supply-chain management modeling. For example, system-theoretic simulation models are being used to evaluate specific supply-chain decision control models specified using linear optimization (Xu and Sen 2005). This distributed computing architecture supports interoperability between discrete-event simulation and linear programming optimization. In this approach, the LP model is wrapped as an atomic DEVS model which allows its execution to be reduced to input and output events. DEVS coupling and simulation protocol are used to ensure correct input and output data exchanges between process and decision models. Input and output transformations have to be carried out inside DEVS and/or optimization models. This approach for process and decision models to interact is limited since the interactions must be divided into simulation and optimization models at a low level of abstraction; thus, timing and synchronization are restricted and ad-hoc. Another key consequence of this approach is partial support for reusability and scalability.

The manufacturing simulation model and decision control model can be specified within a homogeneous environment. For example, a set of manufacturing simulation models and one MPC model were developed in terms of user-defined system blocks and executed as discrete-time models in the SIMULINK/MATLAB environment (Wang et al. 2005). The environment supports synchronous interaction between the simulation models and the MPC model through customized couplings. It is difficult to capture discrete-event behaviors —e.g., modeling unexpected events using discrete-time modeling approaches. Ad-hoc support for modeling discrete event dynamics adversely affects the interactions between process and model predictive control models.

There exist some other combined approaches for simulation-based supply-chain modeling and decision control assessment (Gjerdrum et al. 2001, Venkateswaran and Jones 2004). The interactions between the disparate models are usually held within the model behavioral specification using simulation interoperation and/or software engineering techniques. Very little research has concentrated on studying formal specification of the interactions between system-theoretic simulation modeling and advanced control modeling, although a separate formal interaction model can offer clearer boundaries between distinct modeling formalisms, which are beneficial for composite system model verification and validation.

2 SEMICONDUCTOR MANUFACTURING SUPPLY-CHAIN PROCESS MODELING

Discrete-event simulation has been generally considered suitable for modeling and simulating physical manufacturing process in semiconductor supply-chain systems. Some of its key advantages are handling unexpected events and capturing stochasticity in time and state. Component-based discrete-event modeling approaches can represent the complex manufacturing dynamics at arbitrary levels of detail. For example, a relatively high-level factory model specification can be replaced with by a set of machine models connected together to capture important details such as availability, downtime, and efficiency.

Discrete EVent System specification (DEVS) (Zeigler et al. 2000) is a mathematical modeling formalism for describing (discrete and continuous) dynamical systems as discrete event models—atomic models and/or coupled models.

For semiconductor manufacturing supply-chain systems, there exist a set of primitive process nodes (atomic models)—e.g., *factory*, *warehouse*, *transportation*, and *customer* (Godding et al. 2004, Singh et al. 2004). These nodes can be combined to form complex nodes (coupled models) such as Inventory-Factory (Sarjoughian et al. 2005). The primitive and complex nodes can be connected to constitute a variety of supply chain topologies.

All the supply-chain manufacturing process nodes present common dynamics: (*a*) receive materials (products) from upstream process nodes and release commands from decision models, (*b*) store and/or process materials and process control commands or demands, and (*c*) send materials to downstream process nodes and send status information to the decision model. Given the data and control flows among process models and also with other models such as KIB, it is possible to specify common interface structures for these nodes—i.e., generic input/output ports Data ([Data_In, Data_Out]) and Control ([Control_In, Control_Out]) are used to represent physical material flow and logical information flow, respectively.

// input ports and values
$X = inport \times invalues$
$\quad inport = \{Data\_In, Control\_In\}$
$\quad invalues = \{Lot, Command\}$

// output ports and values
$Y = outport \times outvalues$
$\quad outport = \{Data\_Out, Control\_Out\}$
$\quad outvalues = \{Lot, Status\}$

// state sets
$S = phase \times \sigma \times \overline{Q}$
$phase = \left\{ \begin{array}{c} Initialize, WaitForDecision, \\ StartMaterial, Process, UpdateStatus \end{array} \right\}$
$\sigma : \Re_{0,\infty}^{+}$
$\overline{Q} : Q_{input} \times Q_{storage} \times Q_{output}$

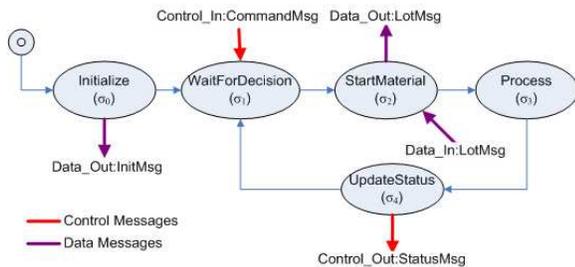Figure 1: Partial Factory Model Specification

Figure 2: Factory Model State Chart

Each type of process node offers specialties as well. A warehouse node can hold material and allow material to leave according to release commands. A key piece of information for a warehouse node is its inventory level—i.e., *beginning-on-hand*, *BOH*. A factory model can represent daily manufacturing operations such as building, assembling, testing, and splitting products, or some combinations thereof. A factory model can have *capacity* as well as stochastic *yield* and *throughput time (TPT)*. The state information that needs to be considered includes *WIP* (*work-in-progress*), *AO* (*actual-out*), average *TPT* and *yield*. A transportation node represents transportation delays; it can be treated as a simplified factory model that cannot change products. A customer node can send demands to both decision model and process model and receive product orders from upstream warehouses. The detailed DEVS specification of a factory model is described as follows.

The structural specification of the atomic factory model is defined in Figure 1 (Godding et al. 2004, Singh et al. 2004). The state set includes two generic state variables *Phase* and *sigma* (σ) and domain-specific state variables for capturing behavior of the factory. For the factory model described here, $\sigma_0 = 0, \sigma_1 = \sigma_2 = \sigma_4 = 0.1$, and $\sigma_3 = 0.7$ (see Figure 2).

A partial statechart depicting external and internal events of the *factory* model is shown in Figure 2. External events for a factory model can be from other manufacturing nodes (e.g., inventory) or external models. External events from a source other than manufacturing nodes (e.g., KIB) are supported with special Control ports. These Control and Data ports play a key role in separating a DEVS manufacturing model interaction with a non-DEVS model (e.g., computational entity).

The processing procedure of each *phase* can be informally described in the following: *1)* in the phase *Initialize*, a set of model parameters are configured—e.g., capacity, normal yield, or yield distribution, *2)* commands, if there are any from the decision control model, arrive as external messages through Control_In port in phase *WaitForDecision*, *3)* at the end of phase *StartMaterial*, the lots in Q_{output} are sent out to the downstream process models through Data_Out port; at the same time, the lots from upstream process models may arrive as external messages through Data_In port; they are initially held in Q_{input}, *4)* during the phase *Process*, lots in Q_{input} are first moved into $Q_{storage}$ and then the lots in $Q_{storage}$ are processed (e.g., lots' processed-time is increased and/or product names are changed); those lots which have been completely processed are then transferred to Q_{output}, and *5)* At the end of phase *UpdateStatus*, status messages (e.g., WIP) are sent out to the decision control model through Control_Out port; meanwhile, the factory's local capacity is sent out to its immediate upstream inventory. A daily operation cycle is completed. The model goes back to step 2 for a new operation cycle.

Table 1: TPT-Load Model

Cases	Load (%)	TPT (time unit)		
		MIN	AVE	MAX
3 levels of distribution	(0 − 70]	30	32	34
	(70 − 90]	32	35	38
	(90 − 100]	35	40	45
5 levels of distribution	(0 − 70]	30	32	34
	(70 − 80]	31	34	36
	(80 − 90]	32	35	38
	(90 − 95]	34	37	42
	(95 − 100]	36	40	45

The lot processing needs to represent realistic manufacturing processes. It must account for not only various processing operations but also stochastic characteristics. In a more complicated circumstance, the uncertain behavior can depend on one or more temporal properties. In particular, the TPT may vary depending on the run-time load of the factory—i.e., heavier load results in longer throughput time. The TPT-load relationship may be specified in discrete or continuous forms given the source of data or assumptions made on non-linearity and stochastic duration (time) for each factory operation.

Modeling of the TPT-load computational relationship play a significant role in representing practical manufacturing processes, since linear or exponential relationships can lead to vastly different supply-chain dynamics. One simplified approach is that TPT can be divided into two or more ranges given different percentages for factory loads. Within specific load percentages, the TPT can be uniformly or triangularly distributed between specified values (see Table 1).

3 PREDICTIVE OPTIMIZATION-BASED PROCESS CONTROL MODELING

Model Predictive Control (MPC) is aimed at controlling dynamical systems using a combination of control theoretic and optimization techniques (Qin and Badgwell 2003). It has been shown that MPC can be used as a tactical controller for high volume semiconductor supply-chain systems to handle nonlinear, stochastic dynamics and unpredictable demand changes while enforcing constraints on desired inventory levels and production and transportation capacities (Wang et al. 2005). In the MPC model, a system prediction model, which is a simplified system as compared with the real system or a detailed simulation model as described in the previous section, estimates future inventory levels by considering historical information such as warehouse inventories, actual customer demands and factory release starts, reference points (i.e., inventory targets), and forecasting customer demands. In the optimizer, a sequence of factory starts is calculated by solving the optimization whose objective functions include keeping inventory levels close to targets, minimizing changes in the manipulated factory starts, and maintaining the factory starts at strategic planning targets.

To meet the requirements of the semiconductor supply-chain tactical control, the MPC model offers different tuning parameters defined for filters or controllers (e.g., Kalman filter)(Wang et al. 2005). These parameter values are key in achieving desired performance requirements for forecasted demands, inventory targets tracking, and unforecasted customer demands. For example, a tuning parameter $\alpha_j \in [0,1), j = 1,2,...,n$ represents each inventory target tracking speed—smaller can result in a faster response that an inventory can achieve to track the target. A tunable parameter filter gain (f_a) is used to deal with prediction errors generated by the stochasticity and uncertainty in the system. As f_a approaches zero, the controller ignores most of the prediction errors and the solution is mainly determined by the deterministic information (e.g., normal system delays and measured anticipations). In contrast, the controller will compensate all of the prediction errors from the stochasticity and uncertainty if f_a equals one.

4 SIMULATION/OPTIMIZATION ENVIRONMENT

In this work, the DEVS model captures complex dynamics of semiconductor manufacturing processes whereas the MPC model is responsible for tactical control. In industrial settings, these are used complementary to each other for analyzing and solving real-world semiconductor manufacturing supply-chain problems.

The Knowledge Interchange Broker (KIB) for a hybrid DEVS/MPC modeling approach ensures the interactions between two disparate models are modeled and executed correctly (Sarjoughian et al. 2005). The KIB composition specification is an independent model between the process simulation model and the tactical control model. It explicitly describes model interactions in terms of *message transformation*, *concurrency*, *synchronization*, and *timing properties*, which account for both structural and behavioral compositions.

The structural composition is to specify message transformation between two distinct interface structures. For example, in composing models expressed in DEVS and MPC (Sarjoughian et al. 2005) or LP (Godding et al. 2004), the interface structure for a DEVS (atomic and coupled) model consists of a set of input and output ports with message bags to be exchanged with other model components. The interface structure for an MPC model, on the other hand, is a set of variables which can have primitive values (e.g., integers). The modeling language (or environment) may also have an impact on interface structure. In particular, DEVS model specification has complex message types whereas MPC model specification does not. To ensure correct transformation, domain-specific knowledge also needs to be considered, since an application domain (such as semiconductor manufacturing supply-chain) contains its own characteristics such as timing constraints, value ranges, and frequency of information exchanges.

The process simulation model and MPC tactical control models describe different aspects of a system. The process simulation model can provide state information (e.g., WIP and BOH) for each node at the end of an operation cycle (e.g., number of products processed in a factory based on hourly or daily time period). The tactical controller may receive accumulated state information and provide control commands for simulation nodes based on single or multiple time periods. In this circumstance, the message transformation must account for data abstraction consistency (Godding et al. 2004).

In a homogeneous modeling framework, the correct interaction between model components are generally ensured by the individual model behavioral specifications (e.g., time advanced function and internal/external
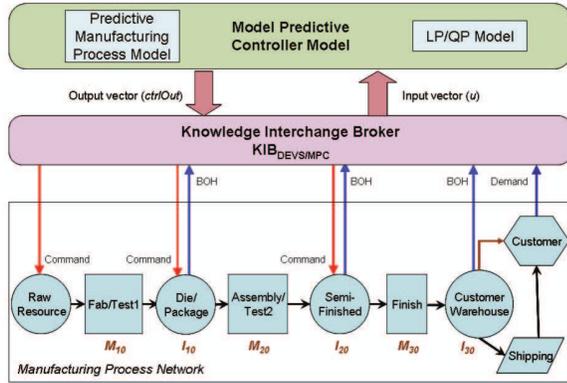
Figure 3: Hybrid DEVS/MPC Model

transition function in an atomic DEVS model) and its well-defined simulation protocol. In a heterogeneous modeling environment, each participating modeling formalism relies upon its own execution protocol to handle timing and synchronization needs. Therefore, it is necessary to synchronize execution of the DEVS simulation protocol and the MPC execution.

To guarantee correctness of execution of models described in DEVS/MPC, the KIB specification is defined as being strictly synchronized (Sarjoughian et al. 2005). The KIB specifies the logical time instance at which the send (or receive) messages must occur between disparate models. In this approach, the wall-clock time consumed by the MPC is abstracted to be instantaneous with respect to the DEVS model and the KIB also does not consume (logical or wall-clock) time.

The time synchronization specification must conform to the synchronization discipline that is associated with the coordination execution control which in turn relies on correct execution of the disparate models throughout the combined DEVS/MPC simulation (i.e., it accounts for causal ordering of messages and time management). The execution of the KIB model generates dynamic behavior of the disparate model interactions and thus allows tracking the data and control exchanges which is essential in validating interactions between the models in a systematic fashion.

5 EXPERIMENTAL RESULTS

The manufacturing process network is limited to contain only one pipeline process which consists of 3 factories (*Fab/Test1*, *Assembly/Test2*, and *Finish*), 4 warehouses (*Raw Resource*, *Die/Package*, *Semi-finished* and *Customer Warehouse*), one *Shipment* and one *Customer* (see Figure 3). These manufacturing model components are described as atomic and coupled DEVS models developed in DEVSJAVA, while the MPC model is the tactical controller implemented in MATLAB. The inter-

action of DEVS model and MPC model is modeled using the $KIB_{DEVS/MPC}$. The experimental testbed also includes some other auxiliary models (e.g., a transducer model is developed for collecting messages).

We have devised a set of experiments to investigate the detailed dynamics of manufacturing process network and to verify the correctness of the composite DEVS/MPC model with $KIB_{DEVS/MPC}$. These experiments help to analyze alternative designs for a prototypical manufacturing process, tactical controller, and their interactions.

Given the above goals, we have devised two categories of experiments. One is to validate the Manufacturing Process Network model with data sets for factory starts and customer demand. The other is to study the MPC model robustness given the detailed simulation of the Manufacturing Process Network model.

In the first category, the factory-starts (i.e., commands to inventories) and customer demand are given to the DEVS simulation model for autonomous process simulation—this enables the simulation model to interact with an idealistic MPC and KIB models. Daily controller commands/demands (defined as standard steps and sinusoidal regime) are sent to inventory *Raw Resource*, *Die/Package*, *Semi-finished* and *Customer Warehouse* to observe stochastic yield and TPT in each factory and validate mass balance among the processing models. The values of upstream factory-starts are set larger than the values at downstream factory-starts given expected yields. Similarly, factory-starts delays are chosen in a such a way to represent realistic dynamics in factory or inventory models. These experiments in this category are important since the manufacturing process model specification will be used in the hybrid DEVS/MPC model.

In the second category, only customer demand is sent to the hybrid system. Sarjoughian et al. 2005 has verified the model composition with $KIB_{DEVS/MPC}$, by assuming relatively simple manufacturing process dynamics. In our experiment, the customer demand uses a square input regime: the average customer demand is set at 951/day with a small variance starting from day 61; the demand increases by about 500 (about 53% percent more than 951) from day 201 to day 400 and then it remains at the average customer demand with a small variance after day 400 until the end of simulation. The aim of this profile is to analyze the robustness of the MPC with respect to sharp increases and decreases in customer demands and thus to determine how well the MPC controller can be tuned to handle unanticipated changes in customer demand, DEVS simulation and $KIB_{DEVS/MPC}$ models.

5.1 Autonomous Process Simulation Analysis

Each node in the Manufacturing Process Network model has detailed dynamics (see Section 2). The factory models can be configured with or without stochastic behavior. Deterministic configuration is used to verify *TPT* and *yield* at each factory and also ensure that mass balance across the entire manufacturing process network is maintained (i.e., the total number of lots entering and exiting the Manufacturing Process Network remains constant). Stochasticity in the factory nodes are modeled by assigning (triangular or uniform) distribution functions to each lot for obtaining actual *yield* and *TPT*. Factory-starts with sinusoidal and square input regimes are sent to each inventory model. The observed information in this experiment includes *WIP* and *AO* for each factory model, *BOH* for each inventory model, and run-time *yield* and *TPT* for each factory model.

It has been shown that, *Lot size*, which is defined as minimal processing unit in every process node, plays an important role since it directly affects a model's stochastic behavior. For example, when the Lot Size is set to a maximum value, the stochastic TPT values generated from the distribution function are assigned to only one lot which includes the amount of all materials flowing in a factory. In this circumstance, the *AO* from a factory may vary significantly (e.g., from 0 in one day to maximum value in the next day). Such sharp changes causing major change in factory load and thus TPT do not reflect realistic behavior of a factory in semiconductor supply-chain system. Appropriate choice of *lot size*, therefore, can significantly impact the dynamics of the manufacturing process model. As expected, the results from a series of experiments with different lot sizes (e.g., maximum, 100, 50, 20 and 10) show that smaller lot sizes give smoother factory behavior as expected. However, a small lot size requires more computation time, which adversely affects the performance of the process and therefore combined DEVS/MPC simulation (see Section 5.4). These experimental results show open-loop DEVS manufacturing process simulations to be logically correct and consistent with prototypical realistic manufacturing systems (Rose 1999).

5.2 Process/OPT Simulation Analysis

The parameter settings for both the process simulation model and MPC model are defined in Table 2. The more-than 50% increase in customer demand can cause significant non-linear dynamics in *Fab/Test1* due to the TPT-load model. It is helpful to know to what extent MPC can handle process nonlinearity and uncertainty.

As shown in Figure 4, more transient states—i.e., inventory level—occur in the *Die/Package* inventory
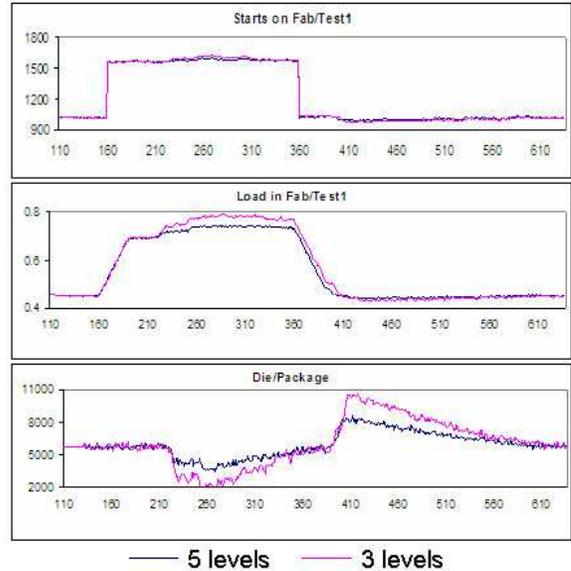

Figure 4: Fab/Test1 Starts & Load and Die/Package Inventory with Different TPT-Load Granularity

models. This was due to the large TPT changes in the upstream *Fab/Test1* factory model. Ideally, when *Fab/Test1* maintains its load within a specific range (e.g., $load \in [72\%, 76\%]$), the average TPT can be kept at 35 days in the process simulation model. Accordingly, such average TPT value is consistent with the corresponding nominal TPT parameter configured in the predictive model inside the MPC model. However, due to the significant increase in customer demand, the start on *Fab/Test1*, which is determined by the MPC model, is increased accordingly to satisfy the customer demand. Consequently, the load in the factory simulation model is also increased. Since the run-time TPT is calculated based on the load, a heavier load can result in longer delay in the *Fab/Test1* model. Long delays in turn impact the inventory level of the downstream *Die/Package* model. Similarly, noticeable transients occur when customer demand decreases sharply.

In contrast to the DEVS simulation model, the nominal TPT configured in the predictive model is deterministic. If the difference between the nominal TPT in the predictive model and the average run-time TPT in the process simulation model is significant, the MPC cannot calculate acceptable factory starts and consequently the DEVS simulation dynamics are not robust. To demonstrate this, another experiment in which a 5-level TPT-load computation relationship is configured in *Fab/Test1* model shows better system dynamics (see Figure 4) since the 5-level TPT-load model has smoother change in TPT in comparison to the 3-level TPT-load model.

The experiments help us analyze and evaluate tactical control policies specified in the MPC model. For

Table 2: Parameter Configurations in Manufacturing Process Network Model and MPC Model

		Process Model								**MPC Model**	
		TPT distribution				Yield distribution (%)				Ave	Ave
		Load (%)	MIN	AVE	MAX	MIN	AVE	MAX	Capacity	TPT	Yield (%)
Factory	*FAB/Test1*	See Table 1				93	95	97	70,000	35	95
	Assembly/Test2	[0,100]	5	6	7	98	98.5	99	10,000	6	98.5
	Finish	[0,100]	1	2	3	98.5	99	99.5	5,000	2	99
	Shipping	[0,100]	1	1	1	100	100	100	2,500	1	100
Inventory		Maximum Capacity								Target Point	
	Die/Package	20,000								5,712	
	Semi-Finished	10,000								2,856	
	Customer Warehouse	10,000								1,787	
		Lot Size				Simulation time				f_a	
Others		50				638 days				0.01	0.05

example, based on the analysis of the experimental results, one of the future research areas is to develop an adaptive MPC model in which it can support dynamic nominal TPT on the basis of certain criteria.

Given the experiments described above, apparently small changes in either the process simulation model or the MPC model can cause significant changes in the manufacturing supply-chain system dynamics. The hybrid discrete-event simulation with optimization control makes it convenient for us to detail and extend manufacturing process simulation and tactical control models separately. The separation gives us a better understanding of both of the models and their interactions. The component-based modeling and simulation environment supports model reusability and configuration flexibility, which also simplifies setting up different experimentation scenarios.

5.3 Execution Time vs. Accuracy Analysis

Execution time for simulation studies depend on a variety of a factors including details of models, efficiency of individual components of the DEVS/MPC environment (DEVSJAVA, SIMULINK/MATLAB and KIB) and the underlying computing environment(Java Runtime Environment), and computer operating system and hardware configuration. For example, in the above experiments, we have chosen *lot size = 50* since it provides a suitable trade-off between accuracy and performance. For example, the DEVSJAVA simulation time can be reduced by about 30% when changing lot size from 10 to 50 (see Figure 5) while maintaining acceptable accuracy of the combined DEVS/MPC models. With this testbed, we can measure and compare wallclock execution time for DEVSJAVA and SIMULINK/MATLAB.

Measurement of the execution times helps to understand relative computational resources committed to each component (e.g., DEVSJAVA) and to identify bottlenecks and ways to make them more efficient while ensuring desirable accuracy in simulation results. For
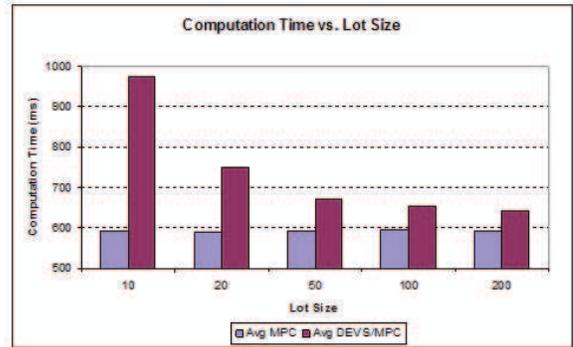

Figure 5: Average DEVS and DEVS/MPC Execution Times

example, in the DEVS/MPC testbed, we have carried out a series of experiments. The results of these experiments are average execution times for a single execution cycle averaged over 5 simulation runs, each of which has 638 cycles—the execution times are shown in Figure 5. One complete execution cycle is measured starting from DEVS to KIB to MPC and back to DEVS. These experiments were conducted on a single computer configured with a 3.2 GHz Intel® Pentium® 4 CPU, 1G RAM and Microsoft® Windows® XP Professional OS, DEVSJAVA 2.7, SIMULINK®/MATLAB® 7.0, and JAVA™ Sun Microsystems JRE 1.5.0.

6 CONCLUSIONS

The experimental results described and discussed in Section 5 show in a quantitative setting the qualitative expectation that higher precision TPT-load models are important in capturing the detailed dynamics between manufacturing processes and decision making and thus help in the modeling of realistic supply-chain systems.

As shown in the experimental analysis, the separation offers flexibility and convenience to explicitly observe and analyze how specific factors on each side can affect the holistic system dynamics. It enforces systematic interactions between disparate models.

ACKNOWLEDGMENTS

REFERENCES

ACIMS 2001. *Arizona Center for Integrative Modeling and Simulation*. Available from <http://www.acims.arizona.edu/> [cited 2006].

Gjerdrum, J., N. Shah, and L. Papageorgiou. 2001. A combined optimization and agent-based approach to supply chain modelling and performance assessment. *Production Planning & Control* 12 (1): 81–88.

Godding, G., H. Sarjoughian, and K. Kempf. 2004. Multi-formalism modeling approach for semiconductor supply/demand networks. In *Proceedings of Winter Simulation Conference*, 232–239. Washington DC, USA.

HLA 2000. *IEEE standard for modeling and simulation (m&s) High Level Architecture (HLA)—Federate Interface Specification*. IEEE.

ILOG 2005. *OPL Studio*. Available from <http://www.ilog.com/products/oplstudio/> [cited 2006].

Kempf, K. 2004. Control-oriented approaches to supply chain management in semiconductor manufacturing. In *Proceedings of IEEE American Control Conference*, 4563–4576. Boston, MA, USA.

Mathworks 2005. *MATLAB/Simulink*. Available from <http://www.mathworks.com> [cited 2006].

Qin, S., and T. Badgwell. 2003. A survey of industrial model predictive control technology. *Control Engineering Practice* 11 (7): 733–764.

Rose, O. 1999. CONLOAD—a new lot release rule for semiconductor wafer fabs. In *Proceedings of the Winter Simulation Conference*, 850–855. Phoenix, AZ, USA.

Sarjoughian, H., D. Huang, W. Wang, D. Rivera, K. Kempf, G. Godding, and H. Mittelmann. 2005. Hybrid discrete event simulation with model predictive control for semiconductor supply-chain manufacturing. In *Proceedings of Winter Simulation Conference*, 256–266. Orlando, FL, USA.

Sarjoughian, H., and J. Plummer. 2002. Design and implementation of a bridge between RAP and DEVS. Computer Science and Engineering, Arizona State University, Tempe, AZ.

Shapiro, J. 2001. *Modeling the supply chain*. Duxbury.

Singh, R., H. Sarjoughian, and G. Godding. 2004. Design of scalable simulation models for semiconductor manufacturing processes. In *Proceedings of the Summer Computer Simulation Conference*, 235–240. San Jose, CA, USA.

Tsai, W., C. Fan, and Y. Chen. 2006. DDSOS: a dynamic distributed service-oriented simulation framework. In *39th Annual Simulation Symposium*, 160–167. Huntsville, AL, USA.

Venkateswaran, J., and A. Jones. 2004. Hierarchical production planning using a hybrid system dynamic-discrete event simulation architecture. In *Proceedings of the Winter Simulation Conference*, 1094 – 1102. Washington D.C., USA.

Wang, W., D. Rivera, and K. Kempf. 2005. A novel model predictive control algorithm for supply chain management in semiconductor manufacturing. In *American Control Conference*, 208–213. Portland, OR, USA.

Xu, Y., and S. Sen. 2005. A distributed computing architecture for simulation and optimization. In *Proceedings of the Winter Simulation Conference*, 365–373. Orlando, FL, USA.

Zeigler, B. 2006. Embedding DEVS&DESS in DEVS. In *DEVS Integrative Modeling & Simulation Symposium*, 125–132. Huntsville, AL, USA.

Zeigler, B., H. Praehofer, and T. Kim. 2000. *Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems*. 2nd ed. Academic Press.

AUTHOR BIOGRAPHIES

DONPING HUANG is a PhD candidate in the Computer Science and Engineering department at ASU. She can be contacted at <dongping.huang@asu.edu>.

HESSAM S. SARJOUGHIAN is Assistant Professor of Computer Science and Engineering at ASU. He can be contacted at <sarjoughian@asu.edu>.

DANIEL E. RIVERA is Associate Professor Chemical and Materials Engineering, ASU. He can be contacted at <daniel.rivera@asu.edu>.

GARY W. GODDING is a Technologist at Intel Corporation and a PhD candidate in the Computer Science and Engineering department at ASU. He can be contacted at <gary.godding@intel.com>.

KARL G. KEMPF is Director of Decision Technologies at Intel Corporation and Adjunct Professor at ASU. He can be contacted at <karl.g.kempf@intel.com>.