Simulation of Semiconductor Manufacturing Supply-Chain Systems with DEVS, MPC, and KIB

Dongping Huang, Hessam Sarjoughian[1], *Member, IEEE* Wenlin Wang, *Member, IEEE* Gary Godding,
Daniel Rivera, *Senior Member, IEEE* Karl Kempf, *Member, IEEE*, and Hans Mittelmann

Abstract

The dynamics of high-volume, discrete-parts semiconductor manufacturing supply-chain systems can be described using a combination of Discrete EVent System Specification (DEVS) and Model Predictive Control (MPC) modeling approaches. To formulate the interactions between the discrete process model and its controller, another model called Knowledge Interchange Broker (KIB) is used. A robust and scalable testbed supporting DEVS-based manufacturing process modeling, MPC-based controller design, and the KIB$_{DEVS/MPC}$ interaction model is developed. A suite of experiments have been devised and simulated using this testbed. The flexibility of this approach for modeling, simulating, and evaluating stochastic discrete process models under alternative control schemes is detailed. The testbed illustrates the benefits and challenges associated with developing and using realistic manufacturing process models and process control policies. The simulation environment demonstrates the importance of explicitly defining and exposing the interactions between the manufacturing and control subsystems of complex semiconductor supply-chain systems.

Index Terms

Discrete Event System Specification, Hybrid Simulation Testbed, Knowledge Interchange Broker, Model Composability, Model Predictive Control, Optimization, Semiconductor Manufacturing, Supply-Chain Management

I. Introduction

Some best-in-class companies have already achieved 5-6% cost reduction by employing effective supply-chain management (SCM) solutions [1]. With the current scale of international supply-demand networks, a 5-6% difference between near-optimal and non-optimal supply-chain management can be worth hundreds of millions of dollars per year [2]. To achieve even greater efficiency, it is important to better capture the dynamics of supply-chain systems [3], [4]. Indeed, the complexity of supply-chain systems and the difficulties they pose in reducing cost and achieving higher efficiency are widely recognized. A key enabling capability is to develop a robust simulation-based testbed for analyzing and designing the complex interactions taking place inside semiconductor supply-chain systems [5].

A discrete supply-chain system – extending from suppliers through manufacturing and ending with customers – can be viewed to consist of two interacting manufacturing and control subsystems. The semiconductor supply-chain begins with the manufacturing subsystem, which receives commands and sends its status from/to the control subsystem. For example, an inventory may be commanded to release a number of its inventory holdings to a down stream factory given present variability in the discrete process and future variability in supply and demand. The manufacturing subsystem responds to inventory release commands and sends manufactured products according to factory rules such as maximum inventory holdings. The control subsystem receives material release target goals and process status updates and sends predicted inventory and work-in-progress trajectories to the manufacturing subsystem. The objective of the controller is to support timely, agile responses defined by short-term inventory goals and long-term supply and demand expectations.

Discrete-event simulation models for the manufacturing subsystem are needed. The models must characterize the inherent stochastic, nonlinear dynamics of factories and inventories. Similarly, discrete-time dynamical models

D. Huang, H. Sarjoughian and G. Godding are with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85281; W. Wang is with the IT Manufacturing Systems, Freescale Semiconductor; D. Rivera is with the Department of Chemical Engineering Department, Arizona State University, Tempe, AZ; K. Kempf is with the Decision Technologies, Intel Corporation, Chandler, AZ; and H. Mittelmann is with the Mathematics and Statistics Department, Arizona State University, Tempe, AZ.

[1] Corresponding author.

are required for generating control policies. Given the intricacies involved in manufacturing processes and (short- and long-term) control policies, their *interactions* must be handled in a systematic fashion. Appropriate formulation of interactions between manufacturing and control subsystems should 1) produce accurate assignment of factory capacities that provide the right product at the right time to the intended customer, 2) reduce wasteful product capacity that may later be discarded, and 3) eliminate activities that increase throughput time, such as excessive changes in factory starts and setups, as well as frequent reprioritization of work-in-progress. The result is reduction in manufacturing, shipment, and management costs across the supply-chain enterprise —i.e., generating more revenue and improving customer satisfaction.

While discrete-event process models and optimized control policies are necessary for simulating the overall complexity of supply-chain systems, they are insufficient unless their synthesis is well-defined. To tackle the kinds of complexity that arise in enterprise systems, it is crucial to separately and explicitly model the interactions that take place between the process and control subsystems. To model and simulate manufacturing supply-chain systems, it is important to explicitly model interactions between discrete processes and control policies and have a robust, flexible, and configurable testbed.

To model the interactions between models that are described in disparate modeling formalisms, the Knowledge Interchange Broker (KIB) approach has been introduced [6], [7], [8]. As a multi-formalism model composability framework, input/output mappings, synchronization, concurrency, and timing properties for models can be described at the level of modeling formalisms [9]. The conceptual basis of the KIB is that disparities between different syntax and semantics need to be accounted for with a separate model, thus enabling independent modeling of interactions between composed models.

To have a composable simulation testbed for semiconductor supply-chain systems, a hybrid DEVS/MPC with KIB$_{DEVS/MPC}$ has been developed [8], [10]. This testbed supports composing Discrete Event System Specification (DEVS) [11] and Model Predictive Control (MPC) [12] models. The testbed is implemented using DEVSJAVA [13] as the discrete-event simulation tool and SIMULINK [14] with a MATLAB-QP [15] as the MPC tool. The DEVS/MPC testbed supports efficient execution of DEVS/MPC models and lends itself for validation of complex discrete manufacturing processes. In the remainder of this paper, we present the DEVS/MPC testbed and illustrate the KIB's benefits for developing and evaluating prototypical semiconductor manufacturing supply-chain simulations. Detailed descriptions of DEVS-based manufacturing process models and MPC-based control policies are described in Sections 2 and 3, respectively. The KIB$_{DEVS/MPC}$ model is described in Section 4. The design of the experiments, simulation scenarios, and their evaluations for the above semiconductor manufacturing are described in Section 5. A brief account of closely related work is provided in Section 6. Conclusion and future work are presented in Section 7.

II. DEVS MANUFACTURING MODEL

To describe semiconductor supply-chain manufacturing networks, the manufacturing and assembly processes (factories), the intermediate inventory holding places, the logistics, and the customers must be modeled (see Figure 1 [16], [10]). Raw silicon flows into the fabrication plant, wafers then flow to the assembly process where dies are attached to packages, then product flows to the finish step where final configurations are made, and finally ends with the finished goods being sent to a customer. External instructions into the model specify how much and what type of product to release out of the inventory holding points into the next process step.

The semiconductor network is modeled with four types of entities: *factories*, *inventories*, *shipping*, and *customers*. Factory models can change the physical characteristics of the material flowing through it. Inventory models can hold and release material on command. Shipping can delay arrival of material to the next entity. Customers can generate orders, generate future order forecasts, consume material, and track order fulfillment. The entities are connected with 3 types of flows. Material flows model the actual physical entities flowing through the manufacturing network. Local control models the commands that are sent between entities internal to the simulation. External control models the commands that are generated outside the simulation. The material flow is modeled as discrete lots of wafers or units. A lot contains a batch of wafer or units. Before the assembly test process, a lot can contain up to 25 wafers of material. One wafer contains many dies on a single piece of silicon. At the assembly step the wafers are cut up into individual dies. At that time, lots contain quantities of individual die or units. The size of die unit lots is a configurable parameter to the simulation. Details of the functionality and available states for each of the entities are described next.

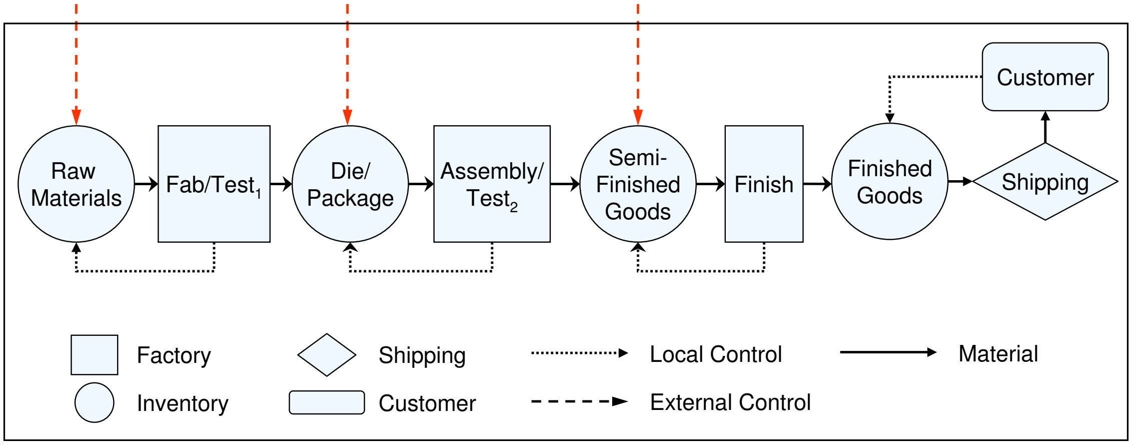

Fig. 1. Semiconductor Manufacturing Process Network

The above semiconductor manufacturing components are modeled using the DEVS modeling framework [11]. In this hierarchical modeling framework, a component is either an atomic model or a coupled model. Each model type has its own input and output ports. Every atomic model has a set of functions that collectively define processing of inputs, changing of states, and generating outputs at arbitrary time instances. A finite number of model components may be connected via coupling their input and output ports which then become coupled models.

A. *Factory Model*

Factory (or process) models change the physical characteristics of material flowing through it. They can model stochastic throughput times and yield loss. A throughput time can be drawn from a distribution and applied to each incoming lot. The random number determines how long it will take the lot to finish the process. Similarly, yield percentages can be drawn from different distributions to determine yield losses or bin splits (e.g. speed or current leakage).

The output states available for external models are work-in-progress (*WIP*) and material left in the last time period. The material leaving is referred to as actual outs (*AO*). The WIP can be reported in buckets. For example, if one configures WIP to be reported in 2 buckets, the output would be two values, the WIP in the front and back halves of the factory. The factory models are able to report their available capacity to other local simulation entities – e.g., see the arrow from the Finish factory to the semi-finished goods inventory shown in Figure 1 [8]. The available capacity is the maximum that can be started into the factory in any given time period.

In our semiconductor model above, there are 3 types of factory models. They are Fab/Test$_1$, the Assembly/Test$_2$, and the Finish. The Fab/Test$_1$ models the facility that fabricates circuitry onto raw silicon and performs the initial testing of the die on wafer. The Fab/Test$_1$ includes both throughput and yield distributions. Lots flowing through this process take varying amounts of time, and the sizes are variable due to the per lot yield losses. The Assembly/Test$_2$ entity characterizes the cutting of wafers, the assembly of die with a package, and the final test. This process consumes a die and a package. A single product flowing into this model can be split into multiple output products depending on the performance distribution measured by the test step. The Finish process sets the final configuration for the assembled material. All factories shown in Figure 1 have stochastic throughput times.

B. *Inventory Model*

The inventory model provides a holding place for material. Material that flows into the inventory will stay there until it is released. Releases can be generated from either external or local control commands. A release command has three parameters, what product to release, how much to release, and where it should be released to. Release messages can be configured to be queued up if they are not fulfilled. In there is not enough inventory to fill a release, or if the output is capacity constrained, the release can happen in the future when the constraining condition does not exist anymore. The inventory can receive a local control message specifying the maximum capacity of the connected entity. For the model in Figure 1, the local control message is used to control how much can be released into the factories. The factory model reports its maximum available capacity to the inventory; the inventory in turn does not allow the maximum to be exceeded. Each of the Raw Materials, Die/Package, Semi Finished Goods, and Finished Goods inventories has two externally available states, the current beginning on hand (*BOH*) inventory, and the actual amount released out (*AO*) in the previous time period. Inventories can hold different kinds and quantities of products.

C. Shipping Model

Shipping models can provide a stochastic delay for materials flowing between entities. The Shipping model is used to characterize air, land, or sea transportation and the associated customs delay. The shipping can have stochastic throughput times and yield losses. Yield losses are used to model damage and theft. The output states available for shipping entities are what is in transit, and how much actually shipped in previous time period. The in transit data can be reported in time buckets similar to WIP for the process model.

D. Customer Model

The Customer model can generate orders, future order forecasts, and track order fulfillment. The current orders are what is currently due and any unfilled orders from the past. The future order forecasts can be output externally as multi-dimensional matrices. For each product the customer has orders for, it can specify a vector of future date/quantity values. Forecast errors can be simulated using distributions. Order fulfillment is tracked by how many orders are filled on time or late. The supply network revenue and customer service levels can be measured from customer logs. For the model in Figure 1, the current actual orders are fed into the finished goods warehouse as release commands. The forecast is sent to the external controller. The controller objective is to manage the inventory releases, getting product to the appropriated holding points in time to maximize customer service levels while minimizing manufacturing costs.

III. MPC CONTROLLER MODEL

Controlling the inherent non-linearity and stochasticity of supply-chain system operations in semiconductor manufacturing is a fundamental goal of this work. This is necessary since a manufacturing network has modes of operations that need to be controlled based on periodic (e.g., hourly/daily) manufacturing process cycles to some other periodic (e.g., weekly/monthly) decision policies in the presence of large and partially unpredictable demand changes. Model Predictive Control (MPC) is a technique arising from the chemical process industries that has been demonstrated to accomplish effective constrained control of uncertain, stochastic multivariable systems through an optimization-based formulation that incorporates both feedback and feedforward decision-making. As a tactical controller for semiconductor manufacturing supply-chain operations, MPC provides robust control and enhanced performance in the presence of significant supply and demand variability and forecasting errors while enforcing constraints on inventory levels and production and transportation capabilities [17], [18]. In the approach described in [17] and [18], a deterministic linear discrete-time model serves as a predictive model for the complex, stochastic, discrete-event model of the manufacturing process. The predictive model has a homomorphic relation to the DEVS process model [8], [10]. The discrete-time factory and inventory models are denoted as $M_{10}, M_{20}, M_{30}, I_{10}, I_{20}$, and I_{30}. For example, the factory responsible for manufacturing products for the “Finished Goods inventory” is modeled as “finish” with its simplified discrete-time model as “M_{30} finish”. The MPC uses the DEVS models to represent the real manufacturing processes (i.e., representing the TPT-load function) and the simplified discrete-time models (i.e., representing a single-value TPT-load as opposed to a multi-value TPT-load [10]) for predicting future inventories which are used by the optimization model. The MPC design in combination with the simplified manufacturing process and detailed optimization must handle stochasticity and uncertainty of the system for some specified time horizon. The optimizer has a set of constraints and an objective function. The predictive (i.e., controller) model is based on the mass conservation relationships among the inventory, factory, and transportation models. For example, the mass conservation relationships for Die/Package inventory level (I_{10}) and Fab/Test$_1$ WIP (M_{10}) are modeled as shown below:

$$\begin{aligned} I_{10}(k+1) &= I_{10}(k) + Y_1\, C_1(k-\theta_1) -\ C_2(k) \\ M_{10}(k+1) &= M_{10}(k) + C_1(k) - C_1(k-\theta_1) \end{aligned}$$

k is an integer variable representing a daily time index; the variable θ_1, meanwhile, represents the nominal (single-value) throughput time, expressed as an integer in days. Y_1 represents the nominal yield for the Fab/Test$_1$ node, while $C_1(k)$ and $C_2(k)$ represent the daily starts sequences that constitute inflow and outflow streams for I_{10} and M_{10}. Similar relationships can be developed for other nodes of the manufacturing process network.

For a given node topology of a semiconductor manufacturing process, the MPC policy manipulates the daily starts of the factories to satisfy the customer demand (both forecasted and unforecasted) while maintaining the inventories at desired levels. The MPC formulation used in this work corresponds to the algorithm developed in [18] and [19]. The general scheme, including its integration with the DEVS simulation model, is described as follows:

1) At initialization, the inventory set-point trajectories are specified. MPC model attributes such as average TPT and yield for each factory model are assigned. The resulting equations for the predictive model are organized into a linear discrete-time state-space model according to:

$$\begin{aligned} x(k) &= A\,x(k-1) + B_u\,u(k-1) + B_d\,d(k-1) + B_{d'}\,d'(k-1) && (1)\\ y(k) &= C\,x(k) + D_{d'}\,d'(k) + v(k) && (2) \end{aligned}$$

The input vectors u, d and d' represent manipulated variables, measured disturbances and unmeasured disturbances, respectively. The manipulated variable vector u physically corresponds to the starts in the manufacturing nodes of the supply-chain, while d represents the forecasted customer demand, which is treated as a disturbance signal with anticipation. y, the vector of measured inventory levels, is the controlled variable, while $v(t)$ is a vector of measurement noise signals. d', the unforecasted demand, is a stochastic signal which can be further described by its own state-space model.

Unlike the discrete-time models used in the MPC model, their corresponding DEVS models have stochastic distribution and have nonlinear TPT. The distribution of the TPT and yield of factory model are assigned at initialization. The TPT and Yield state variables of the DEVS factory models are chosen at the start of the simulation (see Section V).

2) At each subsequent time interval k, the MPC algorithm receives the current inventory levels from the system simulation model. The controller calculations that determine the future starts for each factory involve the following two stages:

- *Prediction:* Relying on the state-space model (2), the controller uses past, current, and forecasted values of inventories, starts, and demand to generate a vector $\mathcal{Y}$ of anticipated inventory levels over a prediction horizon p.

$$\mathcal{Y} = [y(k+1)\ y(k+2) \cdots y(k+p)]^T \tag{3}$$

  The prediction algorithm is structured in this formulation to possess adjustable parameters that moderate the effects of forecasted and unforecasted demand and inventory setpoints.

- *Optimization:* In this stage, a vector of future start changes $\Delta\mathcal{U}$ (also referred to as moves) over a move horizon m is calculated.

$$\Delta\mathcal{U}(k) = [\Delta u(k+1)\ \Delta u(k+2) \cdots \Delta u(k+m)]^T \tag{4}$$

  This is accomplished by solving the optimization problem

$$\min_{\Delta u(k|k) \ldots \Delta u(k+m-1|k)} J \tag{5}$$

  where the individual terms of J correspond to:

$$\begin{aligned} J &= \overbrace{\sum_{\ell=1}^{p} ||Q_e(\ell)(\hat{y}(k+\ell|k) - r(k+\ell))||_2^2}^{\text{Keep Inventories at Inventory Planning Setpoints}} \\ &+ \overbrace{\sum_{\ell=1}^{m} ||Q_{\Delta u}(\ell)(\Delta u(k+\ell-1|k))||_2^2}^{\text{Penalize Changes in Starts}} \end{aligned}$$

$$+ \quad \overbrace{\sum_{\ell=1}^{m} ||Q_u(\ell)(u(k+\ell-1|k) - u_{target}(k+\ell-1|k))||_2^2}^{\text{Maintain Starts at Strategic Planning Targets}}$$

$Q_e, Q_{\Delta u}, Q_u$ are penalty weights on the control error, move size and control signal, respectively; the selection of these weights enables the user to trade-off the ability of the algorithm to satisfy inventory setpoint targets r, adjust starts variability, and maintain starts close to strategic planning targets u_{target} that may be supplied by a higher-level strategic planning module. The problem per (5) can be solved by standard programming algorithms subject to linear inequality constraints. Meaningful constraints in the semiconductor manufacturing supply-chain problem include upper and lower limits on starts, inventories, Work-in-Progress, and their rate-of-change.

3) The starts at time k are sent to the process simulation model. Only the first set of calculated starts in the move horizon are implemented. Each inventory model then releases products to its downstream factory given its local control policy.
4) At the next time interval $k+1$, continue with step 2, using updated information to ultimately compute a new set of future starts over the move horizon. The process of updating information and recomputing an optimal set of decisions corresponds to a *receding horizon* implementation of the Model Predictive Control algorithm.

To better meet the requirements of the semiconductor supply-chain tactical control, the MPC algorithm is devised using a three-degree-of freedom formulation [18], [19]. Such functionality enables the user to independently address the performance requirements associated with meeting forecasted demand (anticipated measured disturbance rejection), inventory targets (setpoint tracking) and unforecasted demand (unmeasured disturbance rejection). The formulation includes adjustable parameters that directly influence the control system response for each performance objective in a manner that is both more intuitive and convenient than accomplished using penalty weights on the objective function. The detailed controller formulation can be found in [18] and [19]. Of particular importance to the results shown in this paper is unmeasured disturbance rejection, which relates to the ability of the control system to respond to unforecasted demand as well as forecast error. The controller's response to these signals is achieved by relying on a specially formulated observer, that recognizes the integrating nature of the supply chain dynamics. The filter acts on the prediction error resulting from the difference between the predicted and measured inventory level values. γ_i corresponds to the adjustable filter gain parameter for each i^{th} inventory, which ranges between 0 and 1. As γ_i approaches zero, the controller increasingly ignores the prediction error and acts in a feedforward-only manner, making decisions on starts primarily on the basis of the forecasted demand signal. The controller attempts to compensate for all of the prediction error from the stochasticity and uncertainty if $\gamma_i = 1$; however, under these circumstances control action can be very aggressive and, consequently, the closed-loop system may be very unrobust. A significant advantage of tuning with γ_i in lieu of the move suppression $Q_{\Delta u}$ is that using γ_i the user is able to influence each output variable independently; move suppression, on the other hand, by penalizing the inputs, consequently affects all of the outputs (albeit in a norm-optimal sense). The tradeoffs associated with the proper selection of the filter parameter in this mode of the controller are described in [19] and further illustrated in Section V of this paper.

IV. KIB$_{DEVS/MPC}$ COMPOSITION MODEL

The DEVS and MPC modeling approaches complement one another and support simulation and analysis of real-world semiconductor supply-chain problems. To develop a model of a semiconductor supply-chain manufacturing network, the concept of Knowledge Interchange Broker (KIB) is used (see Section I). The KIB approach has been developed to formulate the integration of the DEVS and MPC models by modeling their interactions as shown in Figure 2. The KIB Model Specification as a separate model is used to systematically characterize the interactions between the disparate discrete-event manufacturing and optimization-based discrete-time control model. The KIB Execution Algorithm is used to execute the DEVS and MPC execution algorithms. The benefit of model composability and simulation interoperability is that the data and control described in each of the formalisms can maintain their own well-defined syntactic and semantic specifications in a neutral setting.

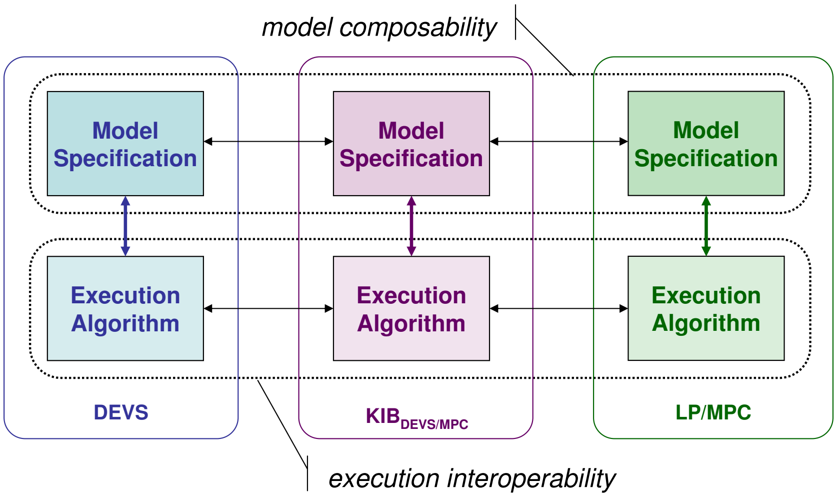

Fig. 2. The KIB Model Composability Approach

Depending on the domain of interest such as semiconductor manufacturing supply-chain, the general modeling constructs of the DEVS and MPC must be accounted for by the KIB in terms of a suite of specific input/output data mappings and transformations. The KIB$_{DEVS/MPC}$ model specification has been developed to ensure the correctness of the integrated structures of the DEVS and MPC models [20]. The KIB$_{DEVS/MPC}$ execution algorithm accounts for the combined execution of the DEVS simulator and the MPC solver in such a way that it can correctly execute the DEVS and MPC model specifications. The composition specification supports simple to complex model interactions that have logically correct structures and can be executed under well-formed protocol. The generality of the DEVS, KIB, and MPC modeling approaches is used to systematically represent the semiconductor supply-chain domain knowledge into the DEVS, KIB, and MPC models. Data mappings and aggregation/disaggregation relationships with synchronized input/output data and control exchanges are handled in a scaleable setting which is key for developing and simulating large-scale Supply-Chain Management (SCM) models.

A model's structure and therefore its interface has an abstract specification so that it can be suitable for different kinds of systems. The DEVS model interface is defined in term of ports and messages. The data contained in messages can have primitive or complex structures. In comparison, the MPC model interface is defined in terms of vectors. The modeling and simulation tools such as DEVSJAVA and MATLAB/SIMULINK allow modelers to specialize generic message and vector types for specific application domains including semiconductor supply-chain manufacturing. The KIB$_{DEVS/MPC}$ is devised to handle structural and behavioral differences between the DEVS and MPC models. A basic concept used in the KIB is to view each model in terms of its external interface and internal execution control. A model's external interface defines what input and output a model can receive and send. A model's internal execution enforces execution of the model according to its semantics. The KIB handles the differences between the DEVS and MPC external interfaces and synchronizing their execution algorithms as described at the end of this section. Next, it is shown how the different kinds of interactions between DEVS and MPC model is handled in a systematic fashion.

The KIB$_{DEVS/MPC}$ specification is defined as a set of nodes (see Figure 3) each of which corresponds to a DEVS model component type as defined in Section 2. Each node specifies bi-directional interactions between a DEVS component model (e.g., Semi-Finished Goods inventory) and the MPC model. Each node model has its own designated mapping and transformation specification. Each node has two responsibilities: (1) mapping the outputs of the DEVS model component to inputs for the MPC model and vice versa and (2) transform the outputs of the MPC model to the inputs for the DEVS model components and vice versa [8], [10], [21], [5]. Consider a synchronous interaction cycle among the DEVS Semi-Finished Goods inventory, KIB Semi-Finished Goods node, and the MPC to have the start and end times of t_k and t_{k+1}, respectively [20]. The Semi-Finished Goods node receives the BOH status message from the DEVS Semi-Finished Goods inventory at t_k and receives the input release command u_i from the MPC before t_{k+1}. The BOH status is sent as output y_i to the MPC at time t_k and the release command message is sent to the DEVS Semi-Finished Goods inventory at t_{k+1}. When a message or a

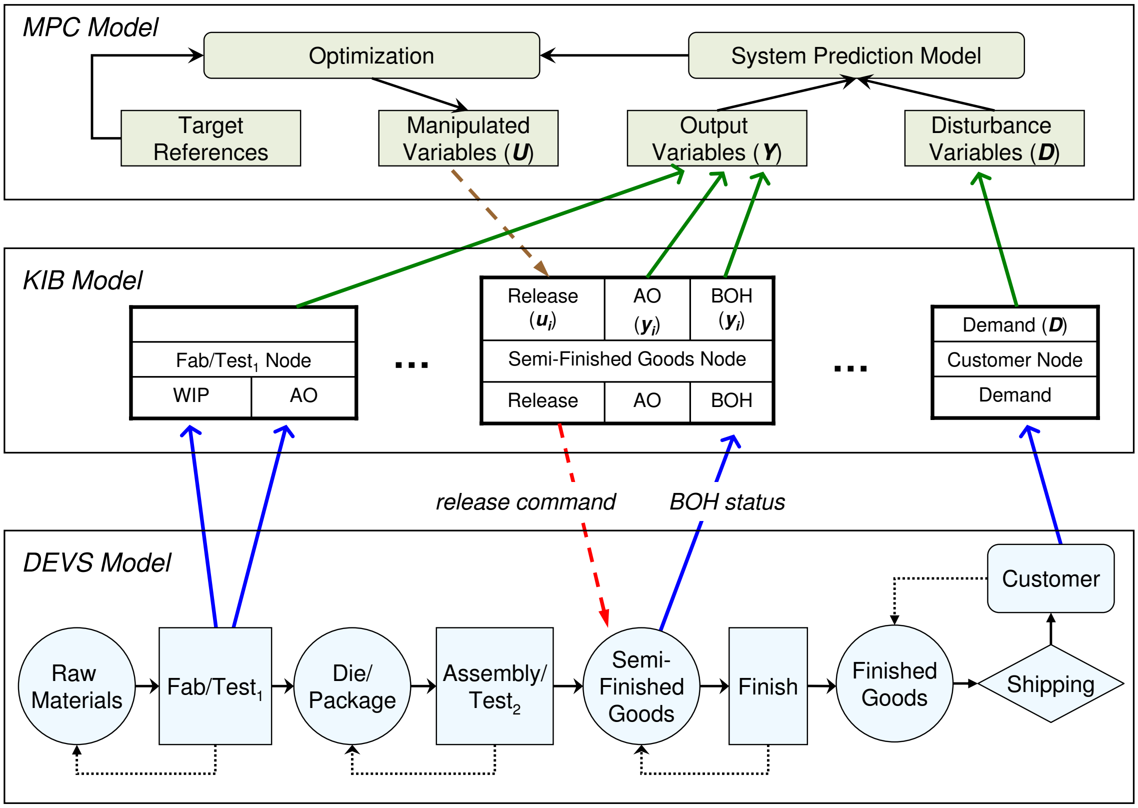

Fig. 3. Composition of the DEVS and MPC models with the KIB model

vector arrives at the KIB, it undergoes a suitable mapping based on the DEVS message type and MPC data type. In general a mapping from DEVS to MPC results in a DEVS message port to be discarded and a mapping from MPC to DEVS results in adding port which is necessary for creating a DEVS message [8]. Removal and addition of port information accounts for inherent disparity between DEVS and MPC modeling formalisms. Aside from mapping, the data content of a message or a vector can be transformed as desired (e.g., partitioning the MPC manipulated variable into individual inventory release commands categorized based on the type of material, destination, and quantity [5], [22]). The data transformation for the KIB Semi-Finished Goods node has the following specification (see KIB Model in Figure 3).

$BOH(material, quanity) \longrightarrow y_i$ where $BOH(material, quanity)$ is the message content received from $\text{DEVS}_{\text{Semi-Finished Goods}}$ model component and y_i is the i^{th} data input for the MPC model,

$u_i \longrightarrow Release(material, destination, quantity)$ where u_i is the i^{th} release command received from the MPC model and $release(material, destination, quanity)$ is the input message content for the $\text{DEVS}_{\text{Semi-Finished Goods}}$ model component.

An inventory model such as Semi-Finished Goods holds one or more material types. The DEVS inventory model component sends its status (e.g., BOH) as output or receive release command as input for current time t_k or for a finite future time period ($t_{k+m}, m \in N$). The status output is a collection of different kinds of material lots and each is associated with one time instance. This output has the same structure as in any other inventory model (e.g., Die/Package). The KIB is specified to support the DEVS Inventory Model interface in general and specialized systematically for different kinds of inventories [16]. Similarly, the release command output generated by MPC is a vector of release commands for different types of material lots. The difference among the release commands is that the time associated with the lots to be released into factories can be the current execution time or a future time instance. The input and output messages for the $\text{DEVS}_{\text{Semi-Finished Goods}}$ (i.e., $release\ command$ and $BOH\ status$) are defined as follows with respect to the $Release$ and BOH structures defined for the KIB's Semi-Finished Goods node model (refer to Figure 3).

$release\ command = \sum_p^m (material_p, destination_p, quantity_p, t_m)$ where different products ranging from $1, \cdots, p$ are released to the Finish factory at future time steps ranging from $1, \cdots, m$ from current time t,

$BOH\ status = \sum_p (material_p, quantity_p)$ where BOH is the sum of different products ranging from $1, \cdots, p$ in the Semi-Finished Goods inventory.

The $Release$ output data content is used to create input message $release\ command$ for the model component. Similarly, the BOH data content is used to store the data content of the $BOH\ status$ that is received from the $\text{DEVS}_{\text{Semi-Finished Goods}}$ model component. Given the difference between the input/output of the DEVS and MPC models, the Inventory Model node maps and aggregate status of the DEVS inventory model and disaggregate the release command of the MPC model [10], [5]. The synchronous interaction cycle between the DEVS process model and the MPC controller follows the following steps. The execution cycle consists of an initial step that initializes the variables that hold the input and output events and numerical values for the semiconductor manufacturing process model and the controller. The specifications of the factory, shipping, and customer nodes follow the principles that are used for the Inventory Model node.

1) DEVS process model computes status events (messages) and sends them to the KIB,
2) KIB transforms and maps the status events to numerical status values (vectors),
3) KIB sends the numerical status values to the MPC controller,
4) MPC tactical controller computes the command numerical values and sends them to the KIB,
5) KIB transforms and maps the command numerical values to command events, and
6) KIB sends the command events to the DEVS process model.

V. EXPERIMENTS

In this section, a set of simulation experiments are described to show the dynamics of the interacting DEVS and MPC models with the $\text{KIB}_{DEVS/MPC}$ model. The correctness of the DEVS/MPC testbed for prototypical semiconductor supply-chain manufacturing (see Figure 3) with an optimization-based controller as well as their interactions have already been shown [10], [20]. Experiments were devised and simulated with predetermined inventory release commands and customer demand. Daily controller commands are defined as standard step and sinusoidal regimes. These factory starts correspond to idealistic customer demand which shows that the DEVS models correctly represent the fundamental dynamics of realistic semiconductor supply-chain manufacturing. Customer demand profile with the average demand was set at 951 with a small variance between 939 and 968 starting from day 61 until the end of the simulation.

TABLE I
TPT-LOAD MODEL

Cases	Load	TPT (Day)		
	%	Min	Ave	Max
3-Level Distribution	(0 − 70]	30	32	34
	(70 − 90]	32	35	38
	(90 − 100]	35	40	45
5-Level 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

For modeling and simulation the combination of the DEVS and MPC models, the DEVS/MPC testbed is used [10], [22]. This testbed supports configuring, simulating, and analyzing the DEVS, MPC, and KIB models individually

TABLE II
MANUFACTURING NETWORK MODEL CONFIGURATION

		DEVS Manufacturing Model							
			TPT Distribution (Day)			Yield Distribution (%)			
		Load	Min	Ave	Max	Min	Ave	Max	Capacity
Factory	FAB/Test$_1$	See Table I				93	95	97	70,000
	Assembly/Test$_2$	[0,100]	5	6	7	98	98.5	99	10,000
	Finish	[0,100]	1	2	3	98.5	99	99.5	5,000
	Shipping	[0,100]	1	1	1	100	100	100	2,500
Inventory		Maximum Capacity							
	Die/Package	20,000							
	Semi-Finished Goods	10,000							
	Finished Goods	10,000							
		Lot Size				Simulation Time (Day)			
	Others	50				638			

and collectively. As noted in Section III, MPC relies on a nominal linear model discrete-time model as the basis for its decision-making; the simulations enable the user to assess how well the MPC decision-policy performs when interacting with the nonlinear, stochastic, and discrete-event dynamics of the manufacturing process represented in the DEVS model. The tuning parameters of the MPC, resolution of the DEVS process models (e.g., varying lot sizes), and formulation of the interactions in the KIB$_{DEVS/MPC}$ support systematic experimentation of realistic semiconductor supply-chain manufacturing.

The demand profile considered in the simulations consists of a pulse signal where customer demand increases by 500 units (53% percent variation compared with the average customer demand) from day 201 to day 400 (refer to Figures 4 and 5). This profile examines realistic dynamics of the manufacturing supply-chain and the robustness of the controller given large increase and decrease in customer demand. All simulations are executed for a period of 577 days. 3-level and 5-level TPT-Load configurations for the factory node are given in Table I. A set of parameters that includes yields, capacities, and throughput time distribution for the Assembly/Test$_2$ and Finish nodes are shown in Table II [2], [10], [17]. Factory models are configured with large capacities (i.e., $C_{\text{Fab/Test}_1} = 70,000$, $C_{\text{assembly/Test}_2} = 10,000$, and $C_{\text{Finish}} = 5,000$) to handle large increases in customer demand. The set of nominal parameters for the discrete-time model, the value of the adjustable filter parameter, and target setpoints for the MPC are included in Table III. The *nominal TPT* and *Yield* used by the MPC model are consistent with the *average TPT* and *Yield* for the DEVS manufacturing process model. The *Target Points* define the desired inventory levels in the manufacturing process model. The γ tuning parameter can be configured to control prediction error and deviation from target inventory levels, as described in Section III.

TABLE III
MPC MODEL CONFIGURATION

MPC Model		
Factory	**Nominal TPT**	**Nominal Yield (%)**
Fab/Test1$_1$	35	95
Assembly/Test$_2$	6	98.5
Finish	2	99
Shipping	1	100
Inventory	**Target Points**	
Die/Package	5,721	
Semi-Finished Goods	2,856	
Finished Goods	1,787	
Controller	**Settings**	
γ	0.01	0.05

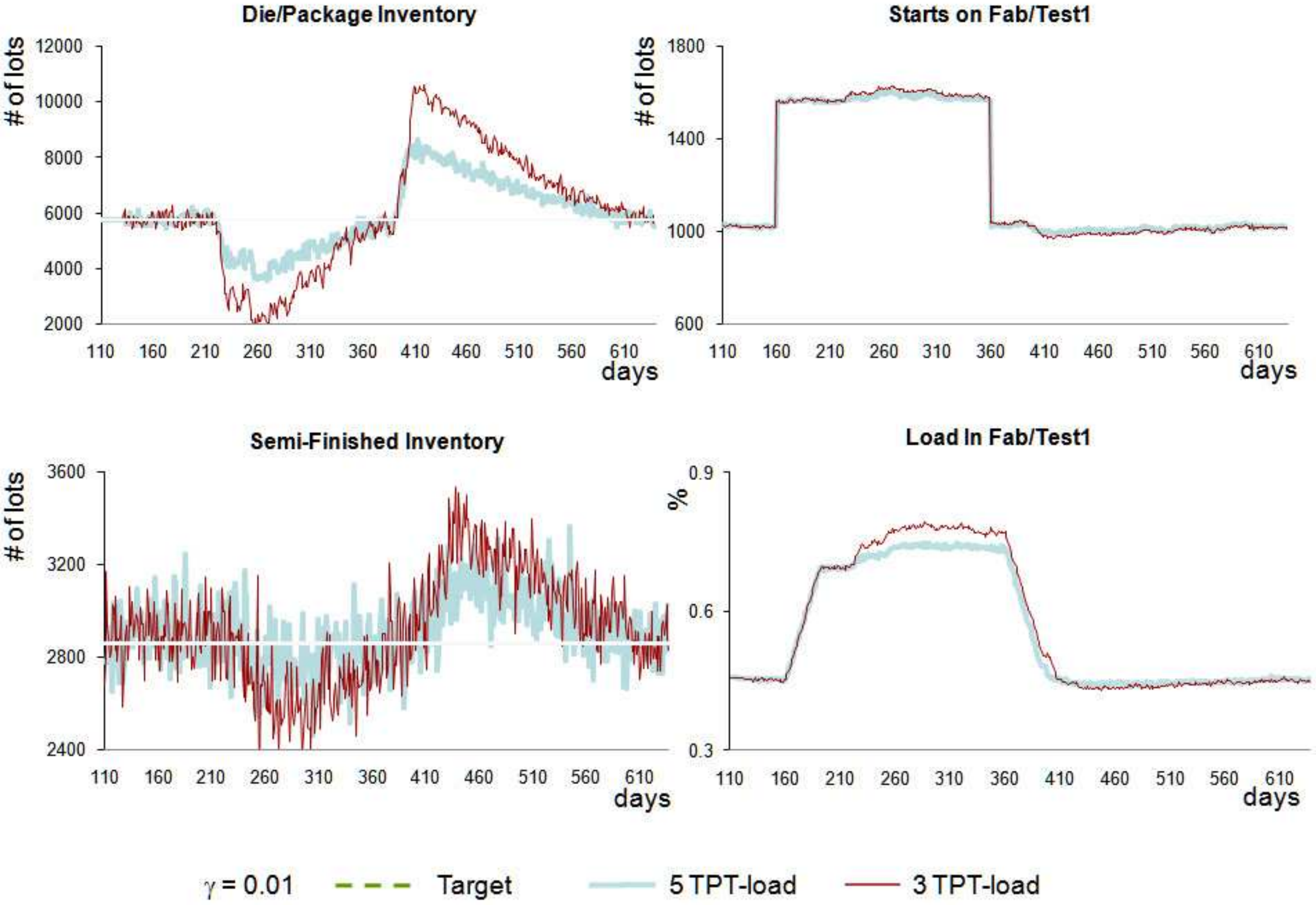

Fig. 4. Effect of varying TPT-load on inventory and factory starts with $\gamma = 0.01$

TABLE IV
INVENTORY VARIATIONS VS. 3-LEVEL AND 5-LEVEL TPT-LOADS

		5 TPT-Load		**3 TPT-Load**	
		Max Difference	**Squared Deviation**	**Max Difference**	**Squared Deviation**
Die/Package	$\gamma = 0.01$	2907	1163.8	4856	2111.7
(Target = 5721)	$\gamma = 0.05$	2374	602.2	3495	1041.1
Semi-Finished Goods	$\gamma = 0.01$	507	155.4	679	237.4
(Target = 2856)	$\gamma = 0.05$	882	276.4	1419	460.8

TABLE V
CUSTOMER BACKORDER

	5 TPT-Load	**3 TPT-Load**
$\gamma = 0.01$	0	144
$\gamma = 0.05$	1037	2710

The first experiment is to evaluate the effect of varying the TPT-load model for a $\gamma = 0.01$. As shown in Figure 4, given a 3-level TPT-load, the Die/Package inventory has substantial transient dynamics due to the significant change in the upstream Fab/Test$_1$ factory model. Ideally, when Fab/Test$_1$ maintains its load within specific range (e.g., ($load \in [72\%, 76\%]$), the average TPT can be kept at the average of 35 days in the process simulation model. Accordingly, such average TPT value is consistent with the corresponding nominal TPT parameter configured in the MPC model. However, due to the significant increase in customer demand, starts on Fab/Test$_1$ are increased. Consequently, the load in the factory model increases. Since the run-time TPT is calculated based on the load,

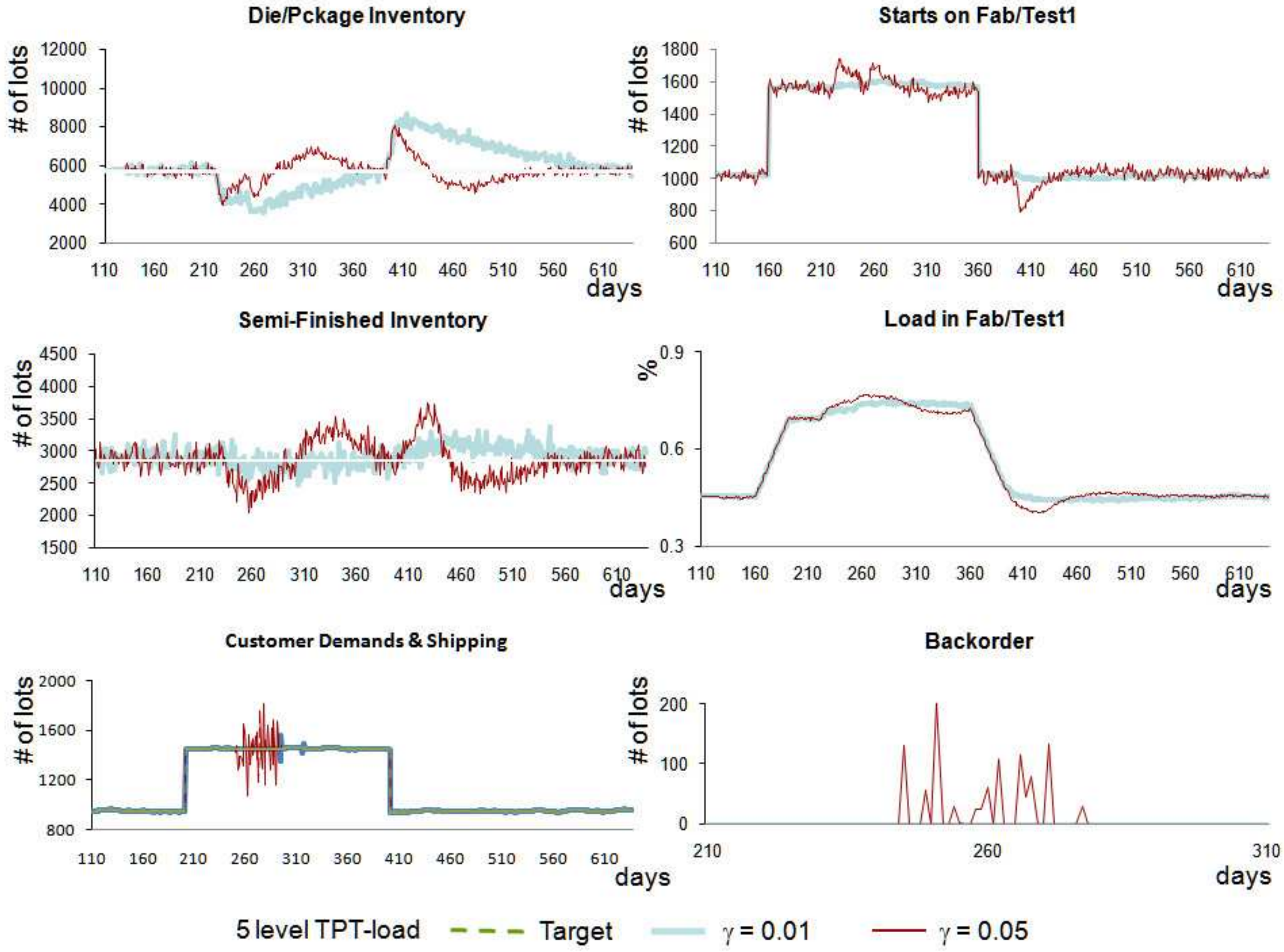

Fig. 5. Effect of varying γ on inventory and factory starts with 5 TPT-load level

a heavier load will cause longer delays in the Fab/Test$_1$ model. Longer delays impact the inventory level of the downstream Die/Package model. Despite the fluctuations, the MPC controller remains stable under conditions of nonlinearity and model mismatch, which is an important practical consideration.

Comparing the 5-level versus the 3-level TPT response (Figure 4) the benefits of the smoother transition between TPT values offered by the 5-level TPT can be seen. Tables IV and V summarize some important characteristics of these responses. The use of the 5-level TPT model results in substantial reductions in both the absolute and sum of squares deviation from target for both the die/package and semi-finished goods inventories. Because the 5-level TPT-load function involves less abrupt TPT changes with respect to load than the 3-level one, one could argue that it represents more realistically the behavior of the factory model, which in turn results in providing more accurate status updates to the MPC decision policy.

The next set of experiments is to evaluate the 5-level TPT-load response for a higher value of $\gamma = 0.05$; the responses for this setting and its comparison with $\gamma = 0.01$ are shown in Figure 5. $\gamma = 0.05$ gives better transient process on Die/Package inventory in terms of fast response and closely tracking targets; from Table IV one can see that both the maximum and squared deviations decrease. The explanation is that compared to $\gamma = 0.01$, a higher percentage of the prediction error is used by the controller to calculate the starts decision to address forecast error. The starts change more aggressively to bring the inventory faster back to the target. A consequence of the more aggressive control action is higher variability on the starts and greater oscillation and deviation from target in the Semi-Finished Goods inventory (Table IV), ultimately leading to backorders (Table V). Based on user preferences the user may choose $\gamma = 0.01$ or evaluate some intermediate γ setting that is less than 0.05.

VI. Related Work

The study of complex systems in terms of modeling their parts and integrating them has been the subject of research across different disciplines and application domains for many years [23]. In the area of manufacturing and semiconductor supply-chain systems, some approaches and testbeds have been proposed for developing strategic plans that can effectively manage complex supply-chain operations [24], [25], [26], [27]. Strategic planning systems employing deterministic LP are useful, but they cannot account directly for the inherent variability in supply and demand. Mathematical optimization and in particular linear programming (LP) optimizers are commonly used [28], [29]. With recent progress in multi-echelon inventory theory, Dynamic Programming optimizers can be used for strategic plan construction [30]. Given target service levels, estimates of future supply and demand uncertainty, and historical forecast bias and error, these inventory algorithms compute safety stock positions and targets to be used as input to the LP optimizers. This safety stock is intended to buffer the expected variability in both supply and demand while executing the LP-generated multi-period plan. These approaches do not support describing the interactions between manufacturing processes and controllers based on model composability approach.

To address the limitation of controlling manufacturing dynamics using decision planning alone, tactical (short-term) control policies, in conjunction with (long-term) decision plans have been shown to handle the stochastic dynamics of manufacturing processes [17]. This approach which is developed in SIMULINK/MATLAB, has been tested under fluid assumptions (discretized continuous-time models). It aims to deal with the supply and demand variability that changes minute to minute, hour by hour, or day after day [2]. The MPC was used with discrete-time manufacturing models to provide fine grain (daily) control, which surpasses the common planning-with-safety-stock approach. The interactions between the simulation model and the MPC were described using the MATLAB macro programming language. With this approach, models of manufacturing processes are discrete-time and thus the role of MPC under the discrete-event setting cannot be evaluated. Furthermore, the integration of manufacturing and control is not grounded in the model composability concepts and methods mentioned above.

Another line of research has employed discrete-event modeling. For example, the manufacturing process and LP models may be modeled using the DEVS framework [31]. With this approach, the LP model must be treated as a DEVS model where customized code is added to the DEVSJAVA software for handle execution of LP solver within the DEVSJAVA simulator. A limitation of this approach is that data and control mappings between DEVS and LP models must be divided between them. Handling complex transformations by dividing data and control mappings between DEVS and LP models is limiting since the modularity cannot be supported. The consequence is reliance on programming and interoperability techniques (e.g., HLA standard [32]) which in turn strongly limits model reuse. Other existing approaches (e.g., [33], [34]) for supply-chain modeling and decision control assessment have similar shortcomings lack of a robust testbed which in turn limits flexible experimentations [10].

VII. CONCLUSIONS

Simulation of semiconductor manufacturing systems requires modeling discrete processes combined with control policies. We have shown the importance of simulating inherently distinct manufacturing processes and control polices using the DEVS/MPC testbed. This novel testbed brings together the complementary DEVS and MPC modeling approaches using the KIB$_{DEVS/MPC}$. A capability of this testbed is independent evaluation of the manufacturing processes, control schemes, and their interactions. The experiments are grounded in a sound hybrid DEVS/MPC modeling framework that supports flexibility for observing and analyzing how discrete-event processes and control policies affect each other. Experiments revealed the impact of realistic non-linearity and stochasticity of the manufacturing dynamics and its importance in designing suitable tuning control parameters. The simulated responses show the ability of the MPC control algorithm based on a linear time-invariant model to maintain stable, robust operation under conditions of nonlinearity and uncertainty in the manufacturing plant dynamics. The simulation studies helped uncover and explain complex relationships between control policies and manufacturing processes. The hybrid DEVS/MPC framework and its testbed are suitable to be extended for distributed simulation and thus supporting large-scale, complex analysis and design of semiconductor supply-chain manufacturing systems. Another area of future research motivated by this study is to develop nonlinear fluid models of the semiconductor manufacturing supply chain as the basis for novel nonlinear MPC controllers. Complementary future research is aimed at developing and simulating manufacturing topologies with multiple controllers.

ACKNOWLEDGMENT

This research is supported in part by grants from NSF Grant No. CMMI-0432439 and Intel Research Council. We would like to acknowledge Kirk Smith of the Intel Corporation and Dieter Armbruster of the Mathematics and Statistics Department at Arizona State University for fruitful discussions about semiconductor supply-chain manufacturing systems. The authors are thankful to the anonymous reviewers whose critiques and suggestions helped improve the presentation and content of the paper.

REFERENCES

[1] ASCET, "Achieving supply chain excellence through technology," `http://www.ascet.com`, 2003.

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

[3] S. D. Wu, M. Erkoc, and S. Karabuk, "Managing capacity in the high-tech industry: A review of the literature," *The Engineering Economists*, vol. 50, pp. 125–158, 2005.

[4] A. Velosa, "Semiconductor manufacturing: Boom busts, and globalization," *The Bridge, National Academy of Engineers*, vol. 35, 2005.

[5] G. Godding, H. Sarjoughian, and K. Kempf, "Application of combined discrete-event simulation and optimization models in semiconductor enterprise manufacturing systems," in *Proceedings of Winter Simulation Conference*, Washington DC, USA, 2007, pp. 1729–1736.

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

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

[8] H. S. Sarjoughian, D. Huang, W. Wang, D. E. Rivera, K. G. Kempf, G. W. Godding, and H. D. Mittelmann, "Hybrid discrete event simulation with model predictive control for semiconductor supply-chain manufacturing," in *Proceedings of the Winter Simulation Conference*, Orlando, FL, USA, 2005, pp. 255–266.

[9] H. S. Sarjoughian, "Model composability," in *Proceedings of the Winter Simulation Conference*, Monterey, CA, 2006, pp. 149–158.

[10] D. Huang, H. S. Sarjoughian, G. W. Godding, D. E. Rivera, and K. G. Kempf, "Flexible experimentation and analysis for hybrid DEVS and MPC models," in *Proceedings of the Winter Simulation Conference*, Monterey, CA, 2006, pp. 1863–1870.

[11] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation*. New York: Academic Press, 2000.

[12] S. Qin, "An overview of industrial model predictive control technology," `http://www.che.utexas.edu/~qin/cpcv/cpcv14.html`, 1996.

[13] ACIMS, *DEVSJAVA*, `http://www.acims.arizona.edu/SOFTWARE`, 2003.

[14] Mathworks, *MATLAB/SIMULINK*, 2005, `http://www.mathworks.com` [cited 2005].

[15] R. J. Vanderbei, "An interior point code for quadratic programming," *Optimization Methods and Software*, vol. 11, pp. 451–484, 1999.

[16] R. Singh, H. S. Sarjoughian, and G. W. Godding, "Design of scalable simulation models for semiconductor manufacturing processes," in *Summer Computer Simulation Conference*. San Jose, CA: SCS, 2004, pp. 235–240.

[17] W. Wang, D. E. Rivera, and K. Kempf, "Model predictive control strategies for supply chain management in semiconductor manufacturing," *International Journal of Production Economics*, vol. 107, pp. 56–77, 2007.

[18] W. Wang, "Model predictive control strategies for supply chain management in semiconductor manufacturing," Ph.D. dissertation, Dept. of Chemical and Materials Engineering, Arizona State University, 2006.

[19] W. Wang and D. E. Rivera, "Model predictive control for tactical decision-making in semiconductor manufacturing supply chain management," *IEEE Transactions on Control Systems Technology*, in press.

[20] H. S. Sarjoughian, D. Huang, G. W. Godding, D. E. Rivera, W. Wang, K. G. Kempf, and H. D. Mittelmann, "Hybrid discrete-event process simulation and model predictive control with knowledge interchange broker," *IEEE Transaction SMC, Part A*, submitted July, 2007.

[21] G. R. Mayer and H. S. Sarjoughian, "Complexities of simulating a hybrid agent-landscape model using multi-formalism composability," in *Proceedings of Agent Directed Simulation, Spring Simulation Multi-conference*, Norfolk, VA, USA, 2007, pp. 161–168.

[22] D. Huang, "Composable modeling and distributed simulation framework for discrete supply-chain systems with predictive control," Ph.D. dissertation, Computer Science and Engineering Department, School of Computing and Informatics, Arizona State University, 2008.

[23] M. Semini, H. Fauske, and J. O. Strandhagen, "Applications of discrete-event simulation to support manufacturing logistics decision-making: A survey," in *Proceedings of Winter Simulation Conference*, Monterey, CA, USA, 2006, pp. 1946–1953.

[24] P. Lendermann, N. Julka, B. Gan, D. Chen, L. McGinnis, and J. McGinnis, "Distributed supply chain simulation as a decision support tool for the semiconductor industry," *ACM Transactions on Modeling and Computer Simulation*, vol. 15, pp. 316–345, 2005.

[25] J. W. Fowler and O. Rose, "Grand challenges in modeling and simulation of complex manufacturing systems," *Simulation Transactions*, vol. 80, no. 9, pp. 469–476, 2004.

[26] J. D. Schwartz, W. Wang, and D. E. Rivera, "Simulation-based optimization of Model Predictive Control policies for inventory management in supply chains," *Automatica*, vol. 42, no. 8, pp. 1311–1320, 2006.

[27] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi, *Designing and Managing the Supply Chain*. New York: McGraw Hill, 2000.

[28] Y.-F. Hung and R. C. Leachman, "A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations," *IEEE Transactions on Semiconductor Manufacturing*, vol. 9, pp. 257–269, 1996.

[29] S. Chopra and P. Meindl, *Supply Chain Management: Strategy, Planning, and Operation*. New Jersey: Prentice-Hall, Upper Saddle River, 2001.

[30] S. C. Graves and S. P. Willems, “Optimizing strategic safety stock placement in supply chains,” *Manufacturing and Service Operations Management*, vol. 2, no. 1, pp. 68–83, 2000.

[31] Y. Xu and S. Sen, “A distributed computing architecture for simulation and optimization,” in *Proceedings of the Winter Simulation Conference*, Orlando, FL, USA, 2005, pp. 365–373.

[32] IEEE, *HLA Framework and Rules*. IEEE, 2000, iEEE 1516-2000.

[33] J. Gjerdrum, N. Shah, and L. Papageorgiou, “A combined optimization and agent-based approach to supply chain modelling and performance assessment,” *Production Planning & Control*, vol. 12, no. 1, pp. 81–88, 2001.

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