Model Composability and Execution across Simulation, Optimization, and Forecast Models

Hessam Sarjoughian[1], James Smith[1], Gary Godding[2], Mohammed Muqsith[1]

[1]Arizona Center for Integrative M&S
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, Arizona, USA
sarjoughian@asu.edu, james.m.smith@asu.edu

[2]Intel Corporation
Chandler, AZ, USA
gary.godding@intel.com

Keywords: DEVS, Demand Forecasting, KIB, Optimization, Semiconductor Supply-Chain, Planning, Simulation.

Abstract

We present a novel simulation platform called Optimization, Simulation, and Forecasting (OSF) for the domain of manufacturing and logistics supply-chain systems. It supports composition of DEVS, Linear Program (LP), and forecast models using an extended Knowledge Interchange Broker (KIB). Models developed in DEVS-Suite simulator, OPL-Studio optimization, and a heuristic Inventory Strategy Forecaster (ISF) can be composed using a new set of scalable XML Schemas developed for DEVS, LP, and ISF models. The addition of forecast modeling offers new kinds of supply-chain system simulation. In particular, alternative customer demand forecast “bias correction” methods can be evaluated towards optimized operation of supply-chain processes. The OSF platform affords modeling interactions among process, optimization, and forecast models. The KIB coordinates simulation (execution) of the DEVS, LP, and ISF models in a sequential fashion. Composition of each pair of DEVS, LP, and ISF models leads to scalability for specifying model interactions. Independent execution of each model allows flexible computation platforms. They simplify defining a large number of different data transformations. The concept, basic architectural design, and implementation of this composable simulation platform are highlighted using example single-echelon and multi-echelon semiconductor manufacturing, logistics systems.

1 Introduction

For a supply-chain system to be efficient across time and space, customer orders have to be placed weeks in advance, factories should be used at their maximum capacity with minimal resources, and sufficient inventories must be available to be shipped in advance of expected delivery dates [3]. This requires a supply-chain system where uncertainty in supplies, process operations, and customer demands can be collectively accounted for. To this end, many people have recognized the importance of both cost effectiveness and high customer satisfaction. To achieve this goal, models for the manufacturing processes and decision making have been previously developed and experimented within an industrial setting. Among existing approaches, discrete-event process simulation using Parallel Discrete Event System Specification (DEVS) and optimized decision making using Linear Programming (LP) have been proposed [2,5]. A DEVS/LP Knowledge Interchange Broker (KIB) has been developed and employed in realistic semiconductor supply-chain application domain [2]. Discrete-event process models simulate physical movement of products, for example, across a semiconductor manufacturing supply-chain consisting of fabrication, testing, inventories, and shipping. Decision plans can project future customer demand using the knowledge of the manufacturing state. Decisions that can lead to (near-) optimal operation of the supply-chain can be generated. The optimization has the task of handling the uncertainty in long-term future customer demand which is inherently inaccurate, sometimes by orders of magnitude. Decisions such as how many products to be moved from one node (e.g., factory) of the supply-chain to another node (e.g., inventory) play a central role – it can increase customer demand satisfaction while lowering cost of operating supply-chain processes.

However, for optimization to generate better decisions, it needs to have reliable customer demand forecasts. Customers project what they need weeks or months ahead of the actual dates the products may actually be needed. This is beneficial for customers as they can overestimate their needs since they often have the option of reducing or canceling their orders without penalty or at greater cost. This results in a “biased customer demand”. In order to handle biased customer demand, forecast modeling has been proposed [3]. A successful bias correction can significantly reduce inaccuracy in customer demand projection. Accurate customer forecast demand modeling such as Inventory Strategy Forecast (ISF) may lend to better optimized decisions, which in turn can increase manufacturing and logistics efficiencies – i.e., better use of manufacturing resources (less cost) while improving customer satisfaction (increased service level).

In the remainder of this paper we will describe a novel simulation platform which integrates three subsystems. This platform is referred to as OSF (Optimization, Simulation, and Forecasting). The DEVS-Suite simulator, OPL-Studio optimization, and their accompanying DEVS/LP Knowledge Interchange Broker already exist [2]. The $KIB_{DEVS/LP}$ supports modeling and executing the interactions between discrete process and optimization models. The third subsystem, an Inventory Strategy Forecaster (ISF), is aimed at countering bias in future customer demand. The addition of this subsystem has resulted in the OSF simulation platform, enabling optimized operations of the supply-chain processes based on handling the uncertainty in future customer demand. The ISF can employ Exponential Smoothing (ES) and Kernel Smoothing (KS) strategies to reduce bias in future customer demand given historical forecast and actual demands. In order to support inclusion of the forecasting subsystems, the $KIB_{DEVS/LP}$ XML Schema is reformulated. The new XML Schema affords scalability in terms of simplifying data exchange definitions as well as composing ISF with both DEVS and LP models. After a brief description of supply-chain process and optimization models, we will discuss some key aspects of ISF. Then, we highlight the KIB reformulation. Thereafter, we present sample results to show the kinds of evaluations that can be supported using OSF platform. In the rest of the paper, we briefly compare this platform with closely related ones and provide a summary of this research and future work.

2 Background

A multi-echelon supply-chain system is a chain of single-echelon (factory or logistic) stages starting from product producers and ending at product consumers. Each echelon is defined to consist of process, shipping, and inventory parts. An example is shown in Figure 1. $Echelon_1$ consists of finished goods warehouse (CW), shipping (Ship), and hub (Hub). These single echelons form a multi-echelon manufacturing, logistics supply-chain where an echelon may have adjoining up-stream and/or down-stream echelons. $Echelon_{i+1}$ is upstream of $Echelon_i$. The most down-stream echelon in a supply-chain of any length is $Echelon_1$. Each element in an echelon can receive inputs, operate on the inputs, and produce outputs. $Echelon_1$ is connected to customers. The customer for any up-stream echelon ($1< i \leq n$) is its immediate down-stream echelon. In this formulation, CW is a process for $Echelon_1$ and a hub is a process for $Echelon_2$. For a supply-chain system spanning multiple geographies, a customer at the end of a supply-chain (i.e., a geographic location for product delivery) is referred to as a Geo-Customer (GC). The products available at the hub can be delivered to GC immediately.

Each part in an echelon can have its own physical characteristics. For example, a finished goods warehouse may have infinite capacity while shipping, hub, and geo-customer may have finite capacity. These parts have other characteristics; in particular, their operations can be controlled externally. For example, flow of products from CW to hub via shipping can be managed using an optimization model that can satisfy a set of constraints defined across an echelon or a chain of echelons. Optimization can determine the release command for CW to Hub – i.e., the number of products that are to be shipped given a desired safety stock at a future time for the Hub. In the multi-echelon case, the optimization model can compute multiple release commands. Release commands can be computed for any number of echelons. Alternatively, in the manufacturing/logistics process, the state of a segment can be used to control the flow of produces within one echelon or multiple echelons. For example, given the states of the finished goods warehouse and hub depicted in Figure 1, the former can compute how many products it can send to the latter [5]. A combination of release commands and state information such as processing time at factory can be used to calculate the number of products that can be sent to the inventory.

Numerous efforts have been undertaken to assist in cost-effective operations of supply-chain systems. A class of approaches and tools use simulation modeling in combination with decision making tools. In particular, simulation can represent past and current state of physical processes (e.g., $Echelon_1$ in Figure 1) whereas algorithms such as linear programming produce plans (i.e., directives) for the manufacturing or logistics process.

The illustration shown in Figure 1 can be concretely characterized from four perspectives. First is physical product engineering and movement (manufacturing/logistics processes). Second is computing optimized plans to operate the physical echelons (optimized processes). Third is removing bias from future customer demand and predicting inventory safety stock (unbiased inventory holdings). Forth is synthesizing interactions among process, optimization, and forecast models using KIB. Details of manufacturing/logistics and optimization models have already been detailed elsewhere [2,5]. In the following section, the ISF from the perspective of OSF will be described and the KIB will be detailed.

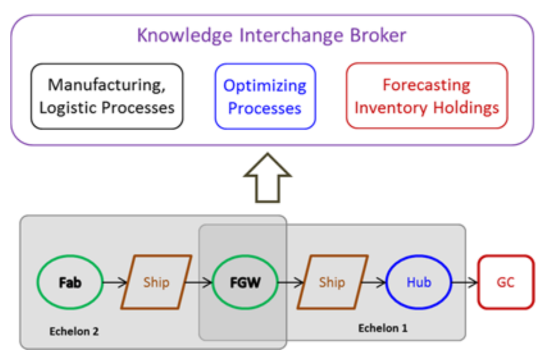


Figure 1. A two-echelon supply-chain system with four kinds of models

3 OSF Platform

The OSF platform consists of optimization, simulation, forecasting, and KIB. Optimization provides a plan for the simulation over a finite time horizon. Simulation captures current state, operations, and events of the semi-conductor manufacturing and logistics processes. Forecasting removes bias from future customer demand. KIB handles data transformations (aggregation and disaggregation) as well as exchange frequency among disparate DEVS, ISF, and LP models. In the following, we describe the novel contributions behind the OSF architecture and two of its constituents (i.e., ISF and KIB).

3.1 Approach

A conceptual architecture for OSF is shown in Figure 2. From a high-level abstraction view, the parts of the diagram show DEVS, ISF, KIB, and LP. Two copies of a database containing Actual Customer Demand (ACD), Historical Forecast Customer Demand (HFC), and Forecast Customer Demand (FCCD) are used with the Simulation and Forecasting parts. Simulation also uses Lot Generator (LG) data contained in the database. Two copies of the database is necessary because ISF is a logical, standalone subsystem both from implementation and execution perspectives (see Section 3.4). ISF can handle multiple echelons based on Base Stock and Multiple Echelon Inventory Optimization (MEIO) approaches [3].

The KIB has three pairs of interfaces (INT), two for each of DEVS, ISF, and LP subsystems. The simulation *state* is sent to optimization and optimization *plan* is sent to simulation. The DEVS and ISF interfaces together can transform simulation state (e.g., hub inventory) and forecast data (e.g., hub safety stock) that are needed for optimization. Similarly, optimization plan can be transformed to meet the needs of the simulation. Basic examples are Beginning On-Hand (BOH) inventories that can be aggregated across seven days, assuming each optimization 'solve' is for a period of one week and simulation execution cycle represents one day. Other exchange configurations can be devised (e.g., optimization, forecasting, and simulation may execute on a daily period or optimization may use the sum of BOHs across multiple inventories). As shown in Figure 2, forecaster has a uni-directional data exchange with optimization. Similarly, simulation and forecasting have a uni-directional relationship where simulation sends its current week number (Wk#) to forecaster. This is needed to guarantee the forecaster is synchronized with simulation and thus creates a well-formed synchronization among DEVS, ISF, and LP model executions. This approach provides simplicity and flexibility as compared, for example, with a design where the forecaster is wrapped inside a DEVS atomic model.

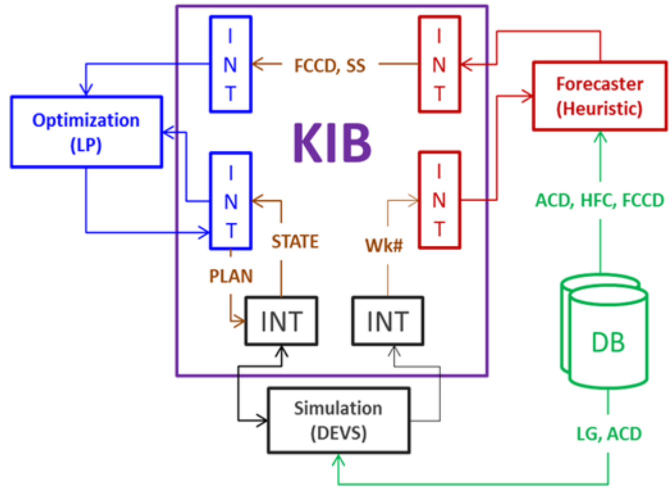


Figure 2. Conceptual OSF architecture

3.2 Inventory Strategy Forecaster

The intent of this section is to highlight the challenge of forecasting inventory holding under demand uncertainty [3,4]. It serves an important role in optimizing the operation of stochastic manufacturing/logistics processes. An ISF, also known as Inventory Strategy Module (ISM) in other works, can employ different bias correction strategies in computing future forecast customer demand and safety stock given historical forecast and actual demands. Excess products can handle unexpected increase in customer demand, but at a cost.

An inventory strategy such as ISF can be used to compute safety stocks to safeguard against inevitable variability in both supply and demand forecasts. Inventory can be characterized in terms of order size, time to order, and holding status. These combined with variables such as future time horizon for which safety target is to be computed can be used to develop different smoothing algorithms. These heuristics algorithms remove bias in future customer demand data given historical and actual data. For this work, an ISF model has been developed using exponential and kernel smoothing techniques [9]. The model can use exponential, kernel, or no smoothing in order to determine safety stock for a future time horizon (e.g., several weeks). Such heuristic models compute minimum safety stocks. It may be possible to minimize manufacturing, delivery, and other costs across supply-chain while maximizing customer satisfaction (i.e., Service Level). In general, a range of safety stock values are computed for a given set of desired service levels (e.g., 60%, …, 100%). The ISF has been developed to demonstrate the OSF platform support for single and two-echelon supply-chain systems. It can handle, in a scalable fashion, multiple hubs and products which are needed for Sequential Base Stock [4] or Multiple Echelon Inventory Optimization (MEIO).

There are well known models for characterizing inventories [4]. An inventory can be characterized to have Bin_1 and Bin_2. The size of Bin_2 specifies the safety stock and the size of Bin_1 specifies the order-up-to. One formulation is called Order-point, Order-Quantity system (s, Q). This model is easy to understand. It can be used when production requirements for supplier can be predicted and no more than one replenishment order is outstanding at any given time. For large transactions, the replenishment may be too small to raise it above the re-order point. Another formulation is called Order-point, Order-Up-to-Level system (s, S). In this model, replenishment quantity is variable. The ISF is defined based on the (s, S) model. In Figure 3, it is shown that the inventory position can be above or below the re-order-point for a given discrete period (e.g., one week). For some on-order and on-hand stocks, inventory position can fall below re-order-point which then requires placing order given order-up-to over some future time horizon (S - IP) and safety stock.

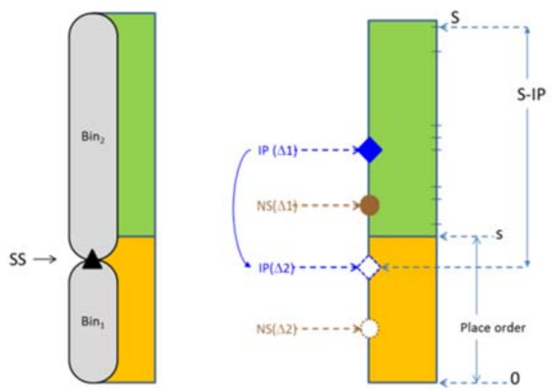


Figure 3. Inventory model

ISF can compute FCCD given HFC, ACD, and the current week (see Figure 4). The kernel and exponential smoothing algorithms reduce (or ideally remove) bias from future customer demand. ISF can then determine safety stocks for the inventories belonging to the manufacturing/logistic system. The optimization model optimizes release commands given safety stock and actual forecast customer demand.

Table 1. Inventory model variables and parameters

Terms	Symbols
inventory position	IP = NS + OO – C
net stock (can be negative)	NS = OH - BO
on-order stock (replenishment)	OO
on-hand stock (stock available in the inventory; cannot be less than zero)	OH
order-up-to	S
re-order-point	s
order-quantity	S-IP
order-quantity (fixed replenishment)	Q
committed (unavailable stock)	C
safety stock (average net stock before replenishment arrival)	SS

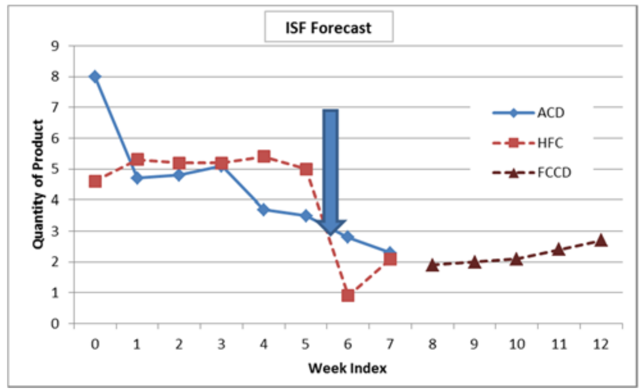


Figure 4. ISF forecast customer demand prediction

3.3 Knowledge Interchange Broker

The abstract KIB model is motivated by multi-formalism modeling [7]. Concrete models, methods, and domain-specific models for semi-conductor supply-chain systems (DEVS/LP [2] and DEVS/MPC [5]) are developed. The basic KIB model is grounded in systems theory where it has inputs, states, outputs, and time-based functions. As an executable model, it has syntax and operational semantics. XML-Schema is used to define syntax of interactions among different kinds of models. For OSF data transformation and execution control, transformation and control schemes are specified as XML constructs that comply with XML-Schemas, defined for DEVS, LP, and ISF (see Figure 5). The use and execution of the data transformations (i.e., XML schema instance models) are achieved using components developed through the Java programming language.

Data transformations are defined and executed according to a clock and time indexes. From the perspective of the simulation, these transformations occur instantaneously. The aggregation, disaggregation, and other input/output mappings occur with respect to the simulation clock and time-indexes contained in data from the models that predict multiple time-steps given the current state of the simulation. For example, LP model computes optimized inventory release commands and holdings for future time-indexes (e.g., several weeks) given the current state of manufacturing and logistics processes. It should be noted that timing aspect of the KIB which is key for time-based data transformations is not shown in Figure 5. The execution control uses clock (current time) and future time-indexes included in the data provided by the ISF and LP models (more details are provided in Section 3.2.2).

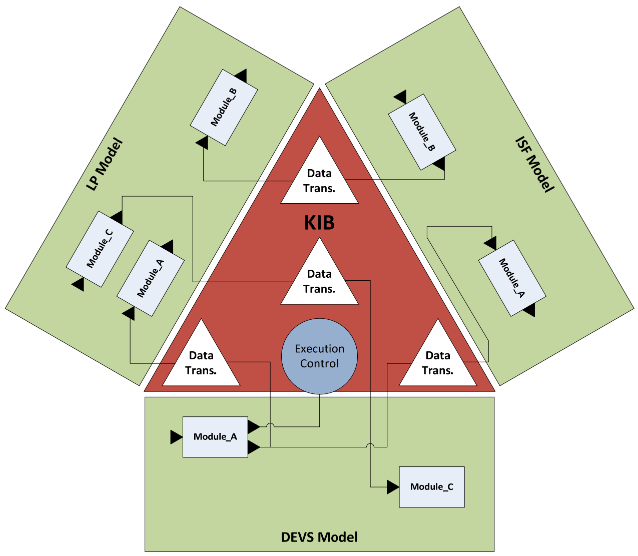
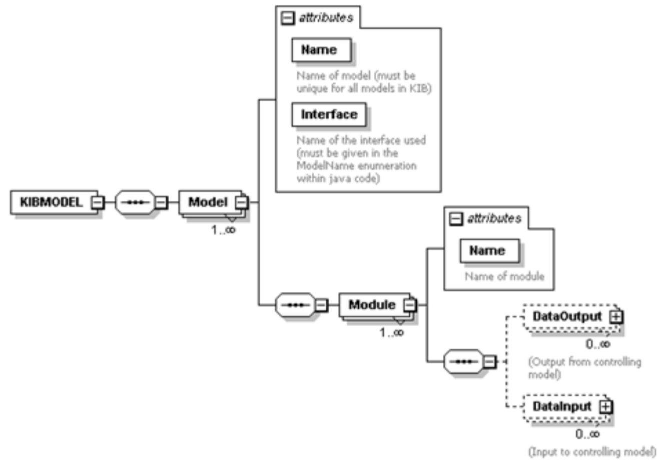


Figure 5. Illustration of the tri-model KIB specification

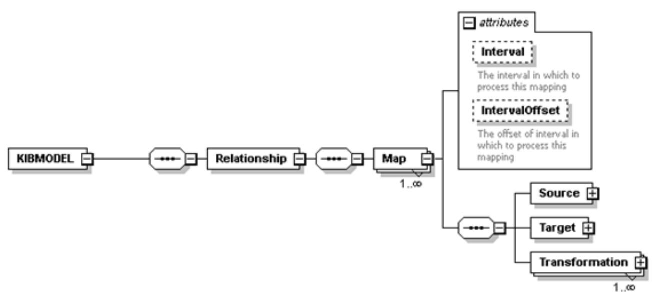
3.3.1 Interaction Schemas

The original XML formulation for data and their transformations were based on a singular view of interactions between two disparate models. An XML instance contained all the elements needed with respect to its counterpart XML instance. Furthermore, elements would refer to the names of the interfaces. This requires redefining XML Schemas where new interfaces are needed. As a consequence, these XML schemas and their instances are weak in terms of generality and scalability.

In order to make the XML Schema more flexible and scalable, a new XML Schema has been developed (see Figure 6) [9]. The new design allows defining data transformations based on the concept of models. A Model XML Schema (see Figure 6(a)) has modules, each of which having input and output data sets. Aside from the Model XML Schema, Relationship (See Figure 6(b)) and Control XML Schemas have also been defined. One is for mapping and transformations. An example partial XML code for ISF/LP is shown in Figure 7. This partial code defines FCCD and hub safety stock outputs for a single-echelon supply-chain system (see Figure 1). In general a single echelon can have multiple hubs with each hub holding different products, and different amount of products may be held at a hub. The data outputs are strongly typed. The echelon index is for multiple echelons (e.g., echelon 1 in Figure 1).



(a) with its constituent modules



(b) Relationships between pairs of source and target models

Figure 6. KIB XML schema specifications

```
<Model Interface="ISF" Name="SupplyChainISF">
  <Module Name="ISF_TARGET">
    <DataOutput Name="FC_CD">
      <DataVariable Name="echelon_index" Type="Int" IsKey="true"/>
      <DataVariable Name="hub" Type="String" IsKey="true"/>
      <DataVariable Name="product" Type="String" IsKey="true"/>
      <DataVariable Name="quantity" Type="Int" IsKey="false"
          ArraySize="Variable"/>
    </DataOutput>
    <DataOutput Name="HUB_SS">
      <DataVariable Name="echelon_index" Type="Int" IsKey="true"/>
      <DataVariable Name="hub" Type="String" IsKey="true"/>
      <DataVariable Name="product" Type="String" IsKey="true"/>
      <DataVariable Name="quantity" Type="Int" IsKey="false
          ArraySize="Variable"/>
    </DataOutput>
      …
</Module>
```

Figure 7. Snippet XML model for ISF model

3.3.2 Execution regime

The executions of DEVS, OPL-Studio, and ISF are independent from one another. Different instances of OSF system models can be developed using the generality of the KIB. An important aspect of this composition is the execution control. Their interactions are controlled in a sequential manner. For example data from ISF is passed through the KIB to LP. Data from the manufacturing/logistics model can be aggregated at the DEVS KIB interface and then delivered to LP. The order of executions is DEVS → KIB → ISF → KIB → LP → KIB [9]. In terms of timing, ISF, KIB, and LP do not consume simulation clock time. This is meaningful in the context of some manufacturing/logistics systems where forecasting, optimization and data transformation can be assumed to be instantaneous.

It can be observed in Figure 8 that ISF is concerned with future whereas the current state of the system as captured in simulation uses cumulative actual customer demand. For a given time *t*, forecast and optimization models predict over n simulation timer periods as compared with simulation which predicts over time period Δt (see Figure 8) For example, simulation is on a daily cycle whereas forecast and optimization are on a weekly basis. When the simulation, forecast, and optimization periods are different, the frequency for data transformation is also specified as part of the KIB execution control.

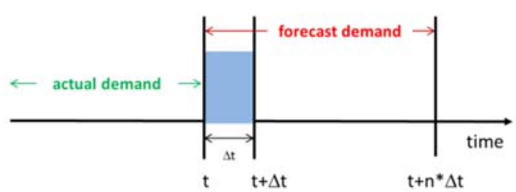


Figure 8. Simulation, forecast and optimization timing with respect to past, current and future time periods

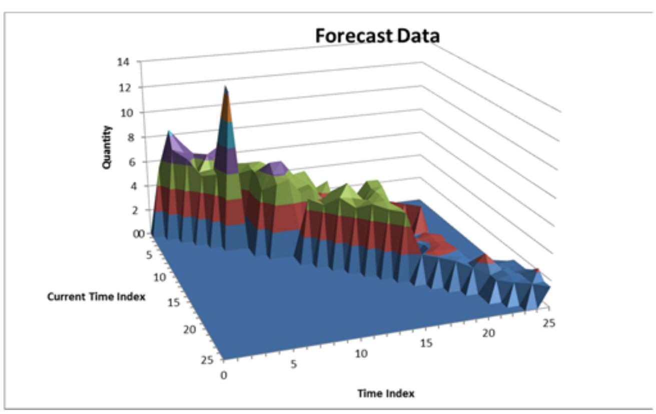


Figure 8. Evolution of forecast demand as a function of current time period

3.4 Implementation

The Optimization, Simulation, and Forecasting (OSF) platform is an extension of the simulation/optimization platform built using DEVS-Suite simulator, OPL-Studio, and $\mathrm{KIB}_{\mathrm{DEVS/LP}}$. In OSF, the latest CPLEX Optimization Studio (version 12.4) is used. The OSF has a Forecaster implemented in Java with RMI support. The KIB is extended to support DEVS, LP, and Forecaster. As briefly described above, the KIB has a new model specification (see Section 3.2) and accompanying execution engine. The simulator, forecaster, and KIB are developed in Java. The optimization IDE tool is also developed in Java with its solver developed in C/C++. The platform supports multiple instances of Inventory Forecasters. This is useful because there are emerging approaches for handling bias in inventory control.

4 Experimentations

The data used for the ISF and more generally for manufacturing/logistics processes and optimizations are for testing the correctness of the OSF design and implementation. The simulation period has its own non-periodic and periodic timing. The forecast and optimization models have their own planning horizon duration and periods for which safety stock and release commands are generated.

Examples from many simulation experiments are shown in Figures 9-11 [9]. The result for single echelon shows the ideal case – forecast model has perfect knowledge of future customer demand. This provides perfect safety stock data for a number of future periods (see Figure 8). The optimization model is able to make perfect release commands given information such as beginning on hand at the finished goods warehouse and having deterministic shipping time. For the first 3 time periods, the customer demand is not met due to initialization – it takes 3 time periods for products from warehouse to reach the geo-customer. The impact of imperfect knowledge of customer demand for the forecast model and stochastic shipping is shown in Figure 9. The importance of exponential, kernel and no smoothing algorithms in calculating safety stocks can be seen in Figure 10. The figure shows the average inventory holdings that are needed to satisfy different customer service levels. The length of a simulation run determines the duration of historical data (ACD, FCCD), planning horizon for forecast and optimization models, and shipping period for a given hub (H1) and product (P4). It is not necessarily true that increasing safety stock holding results better customer service level. For example, calculating safety stock without using any smoothing algorithm (NS) on the future customer demand yields better result than adjusting for demand bias using either exponential or kernel smoothing algorithms.

Experiments with or without bias correction are carried out for the two echelon supply chain system as shown in Figure 1. In this example, it can be seen that inventory holdings for finished goods warehouse and hub are optimized separately. The optimization model can receive safety stocks for these two inventories and compute release commands accordingly. From this configuration, it can be seen that OSF can support each echelon having its own forecast model. This means, for example, no smoothing forecasting can be used for one echelon and exponential smoothing for another echelon. The example average inventory holdings for the finished goods warehouse and hub suggest kernel smoothing may be desired as shown in Figures 11(a) and 11(b).

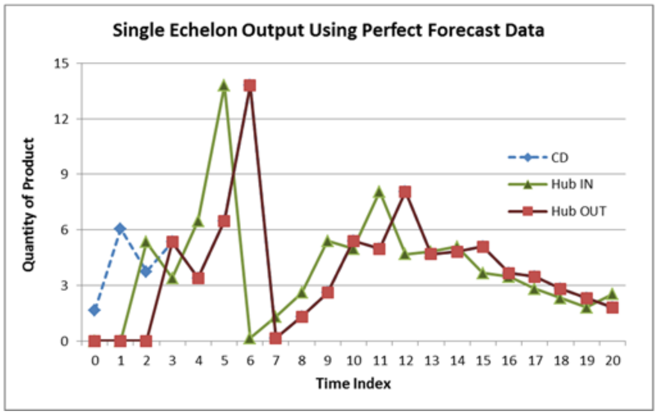


Figure 9. Simulated single-echelon inventory at hub using perfect forecast customer demand

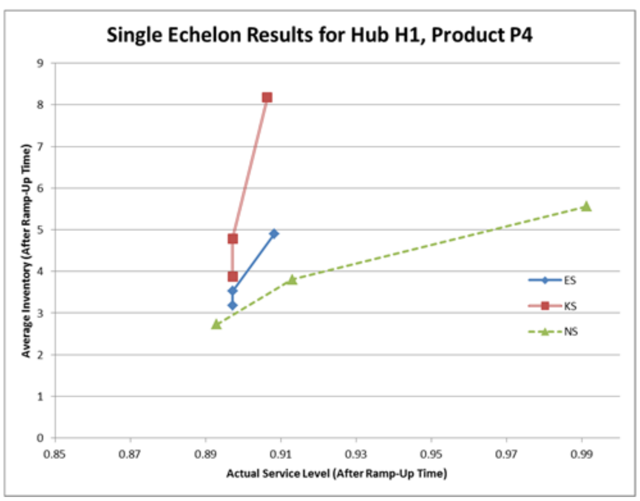
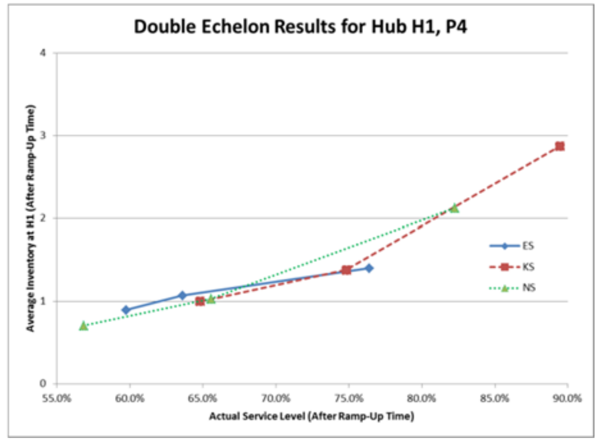
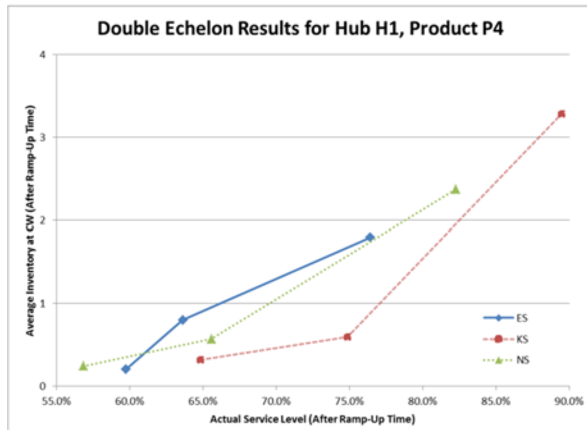


Figure 10. Single echelon with log-normal shipping, 10 day mean, 8 day minimum



(a) Average inventory at H1 (down-stream) echelon



(b) Average Inventory at CW (up-steam) echelon

Figure 11. Inventories across a two-echelon supply-chain system

5 Related work

As described above, this research extends the KIB-based integrated LP optimization and DEVS simulation platform. It introduces scalability for modeling KIB as well as supporting inventory modeling that can counter bias in customer demand forecast. In another research, an "inner and outer loop optimization" is developed for long-term planning and short-term control [10]. It uses Linear and Quadratic Programming (LP and QP) and Model Predictive Control to handle stochasticity due to customer demand in addition to those belonging to manufacturing processes. Optimization works well for coarse grain planning (e.g., days to weeks to months), whereas MPC works well for fine grain control e.g., hourly or daily) of manufacturing processes. To account for variability in customer demand forecast, inventory planning is used. The data from inventory planning is used as external input for LP, QP, and

MPC. This approach is implemented in Simulink/Matlab toolkit. The MPC controller is formulated for a discretized manufacturing process model. Unlike this work, OSF supports explicit modeling interactions between ISF (model inventory) and optimization models.

In another work, control-theoretic techniques commonly used for continuous processes are adapted to discrete manufacturing processes [8]. A control model is developed for a supply-chain system built from production-inventory discrete models. Feedback-feedforward control policy (akin to Proportional–Integral–Derivative controller control) is developed according to the MPC model. This approach has the advantage of responding effectively to short-term changes in inventory targets and changes in customer demand forecast. However, similar to the inner and outer loop optimization approach, the combined process and control cannot provide the kind of model composability flexibility and scalability afforded with the Knowledge Interchange Broker and OSF. Handling complex transformations by dividing data mappings among DEVS, ISF, and LP models provides strong modularity. This contrasts relying on programming and interoperability techniques (e.g., HLA standard) that inherently offers little support for model composability [2,5,6]. These and other approaches also adversely affect flexible design of experiments.

The OSF platform is similar to the previously developed multi-KIB [5] where DEVS, MPC, and LP models can be composed and executed asynchronously. The OSF', a uni-KIB, is defined for the interfaces encompassing DEVS, LP, and ISF models (see Figure 2). The optimization and forecast models execute at an instance of time. This contrasts the multi-KIB approach where DEVS and MPC models execute concurrently. Therefore, the uni-KIB compensability approach has a simpler operational semantics as compared with the multi-KIB.

6 Conclusions

To support study of multi-echelon supply chain systems, a novel Optimization, Simulation, and Forecasting platform has been developed. This platform proposes a Knowledge Interchange Broker for composing DEVS, LP, and ISF models. The KIB has a new, scalable KIB XML schema model. A sequential scheme controls the executions of the DEVS-Suite simulator, CPLEX optimizer, and ISM engine. The platform also introduces forecast modeling which provides basic support for computing safety stocks over multiple hubs and products. Basic example models for single and two echelon supply chain systems have been developed and analyzed.

Experiments using perfect data were conducted to show the correctness of the OSF uni-KIB design and implementation. The platform demonstrates new kinds of interactions among different complex models and can be specified in a scalable and systematic manner. Large input and output data can be transformed in accordance with the modularity of the composed models and the KIB itself. Future work includes developing experimentation concepts and methods that can benefit from interaction modeling. This is expected to be particularly important toward exploring system scenarios that are principally driven by the interactions taking place among simulation, optimization, and forecast models.

Acknowledgement

We would like to acknowledge the contributions of Asima Mishra, David Bayba, and Charles Arnold of the Intel Corporation. A. Mishra formulated the Single-Echelon ISF model. David Bayba and Charles Arnold supported experimentation and data analysis. This project has been funded by Intel Corporation, Chandler, Arizona, USA.

References

[1] DEVS-Suite Simulator, (2009), Simulator, http://devs-suitesim.sourceforge.net/.

[2] Godding, G., (2008), “A Multi-Modeling Approach Using Simulation and Optimization for Supply-Chain Network Systems”, PhD Dissertation, Computer Science and Engineering, Arizona State University, Tempe, AZ, June.

[3] Graves, S.C. and S.P. Willems, (2008), Strategic Inventory Placement in Supply Chains: Non-Stationary Demand,” Manufacturing & Service Operations Management, Vol. 10, No. 2, pp. 278-287.

[4] Hopp, W., M. Spearman, (2011), Factory physics: foundations of manufacturing management, Second Edition, McGraw-Hill.

[5] Huang, D., H.S. Sarjoughian, W. Wang, G. Godding, D. Rivera, K. Kempf, H. Mittelmann, (2009), “Simulation of Semiconductor Manufacturing Supply-Chain Systems with DEVS, MPC, and KIB”, IEEE Transactions on Semiconductor Manufacturing, Vol. 22, No. 1, 165-174.

[6] Kempf, K. G, (2004), “Control-Oriented Approaches to Supply Chain Management in Semiconductor Manufacturing”, Proceedings of American Automatic Control, Boston, MA.

[7] Sarjoughian, H.S., (2006), “Model Composability”, Winter Simulation Conference, Methodology Track, pp. 149-158, December, Monterey, CA.

[8] Schwartz, J. and D. Rivera, (2010), “A Process Control Approach to Tactical Inventory Management in Production-Inventory Systems”, Int. J. Production Economics, Vol. 125, pp. 111–124.

[9] Smith, J., (2012), “Scalable Knowledge Interchange Broker: Design and Implementation for Semiconductor Supply Chain Systems”, MS Thesis, School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, December.

Wang, W., D. Rivera, (2008), Model Predictive Control for Tactical Decision-Making in Semiconductor Manufacturing Supply Chain Management, IEEE Transactions on Control Systems Technology, Vol. 16, No. 5. pp. 841-855.