Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

ADVANCED LOGISTICS ANALYSIS CAPABILITIES ENVIRONMENT

Steven E. Saylor

The Boeing Company
Boeing Research and Technology
Seattle, WA 98124, U.S.A.

James K. Dailey

James Dailey and Associates
Woodbury, MN 55125, U.S.A.

ABSTRACT

This paper presents an approach for modeling supply chain networks and logistics operations using a custom developed supply chain modeling library wholly integrated into a general purpose commercial off the shelf simulation software package. Advantages of the current approach include rapid model development, high degree of reusability, database driven architecture, and the ability to insert prebuilt or custom tailored system operations demand models for assessing the direct linkage of supply chain performance on system operational availability. The current approach has been used to model supply chain management systems, manufacturing and assembly, aircraft fleet availability and performance-based logistics.

1 INTRODUCTION

Operational and support costs of major military and commercial systems typically make up over 70% of customer total ownership costs over the life of the system while development, procurement, and disposal make up less than 30% of total costs (U.S. GAO 2000). Suppliers and customers are increasingly concerned with reducing costs of maintaining and sustaining systems, while at the same time assuring required operational and system availability levels are met. Performance-based logistics support contracts are increasingly replacing traditional transaction based contracts as a means of improving efficiency and reducing overall costs while transferring risk to the service provider contractually responsible for providing a guaranteed availability level. The Boeing Advanced Logistics Analysis Capabilities Environment (BALANCE) modeling simulation architecture was developed to enable accurate, highly user configurable simulations of complex multi-echelon, multi-part indenture supply chain networks, including subassembly and refurbishment. BALANCE is used to support inventory management analysis, performance-based logistics contracts risk assessment, and manufacturing schedule risk assessment.

2 BALANCE ARCHITECTURE

The BALANCE architecture comprises a collection of custom blocks developed in the ExtendSim simulation software language that are used to create supply chain network models and a complex supporting database structure. ExtendSim was selected as the basis for development due to its tightly integrated database architecture (Krahl 2008), underlying messaging architecture (Krahl and Lamperti 1997), and custom block development environment (Krahl 2002) that easily facilitates development of customized database driven simulations. A requirement of the current effort was to develop a standardized reusable supply chain modeling toolset that could be easily configured via database imports and simple graphical user interfaces wherein the user selects options for how to set up and configure policy decisions regarding ordering, shipping, receiving, and refurbishment.

In general, model buildup is accomplished by first populating database tables specifying the composition of the supply chain and associated inventory data parameters, importing that data, then using drag and drop techniques to add prebuilt inventory control point (ICP) modules from the BALANCE library, one for each node in the supply chain network. Then the user selects dropdown menu item settings for setting up the replenishment polices regarding reorder cycles, shipment cycles, order compositions, quantities, supplier selection rules and the like. Alternatively, this data can be pre-populated in the import databases to allow the automatic buildup and configuration of model settings by running a custom autobuilder which has been programmed to read the appropriate database tables, place the appropriate ICP modules onto the model, and configure the associated dialog settings automatically using ExtendSim API calls configured to execute with a simple click on a command button.

Figure 1 shows an example four-tier BALANCE model. In this example squadrons of aircraft are based at forward operating locations and fly missions according to a user defined flight schedule. Aircraft are subject to random part failure and go through unscheduled and schedule maintenance thereby creating demands on the supporting supply chain network. The BALANCE architecture manages the supply chain inventory levels according to defined policies. The ability to keep aircraft flying and mission ready depends upon the performance of the supply chain network to maintain required inventory throughout the network. In this way, we can assess in one comprehensive model the impact of the logistics support concept on individual and overall system availability.

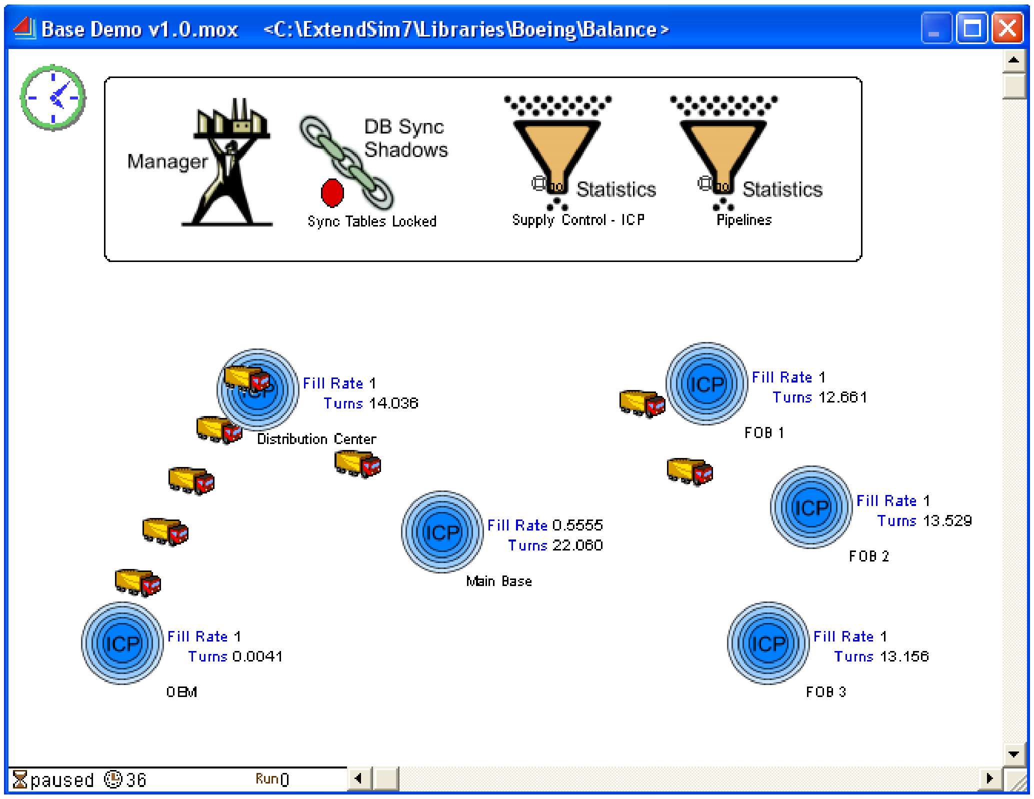


Figure 1: Example BALANCE model

A key feature of the BALANCE architecture is that all data needed to configure and run the simulation is managed and maintained in the underlying database architecture. This includes all dialogs settings or input parameters that may be modified by the user either directly in the appropriate database, or in a given block dialog setting that is then automatically transferred to the database. This provides for complete separation of data from the model, and greatly enhances scalability. For example, macros contained in the manager block housed within an ICP module automatically add a new database to the model each time a new ICP module is dropped onto the face of the model, and registration routines run which then associate that specific ICP module in the model with its own associated database. Moreover the synchronization of data across databases is maintained automatically for the user enabling the model, and more importantly the databases behind the model, to be added to or deleted from or otherwise updated any time a user adds or deletes an inventory control point module, or adjusts a setting within a model. This enables easy scaling of models, either up or down, with minimum burden on the user and enables interfacing with extraction, transform, and load logistics data management systems for the purpose of importing data from a qualified data source. In BALANCE, no entities, or items, are required or generated by the model unless specifically desired by the user to accommodate modeling aspects of the system outside the basic scope of BALANCE, such as details of the system being supporting by the supply chain. This non-item based simulation approach has been implemented by Phelps, Parsons, and Siprelle (2002) in prior modeling methodologies.

2.1 Key Terms

Before we discuss elements of the BALANCE library let us introduce a set of key terms that will be referred to throughout.

Table 1: BALANCE Library Key Terms

Key Term	Description
ICP	Inventory Control Point: A physical place where inventory is held. Maintained as a complete reusable module in the BALANCE library.
Supply	A named item that is held in inventory. Supplies may be an individual unit such as a line replaceable unit (LRU), or consist of one or more sub-units (SRUs), at one or more levels of indenture. All supply items in a BALANCE model are tracked in two states: a fully functional supply that is known as simply "supply" and the same non-functioning supply inventoried as "needs repair".
Requisition	An order or request for supplies. This can be between ICPs or within the same ICP if assembling or refurbishing supplies.
Pipeline	A defined procurement relationship or strategy for replenishing supplies. The pipeline description contains information about what's being requisitioned, from which supplier or group of suppliers, under what conditions, and how frequently. This can include static descriptions that are unvarying during the simulation run, or rule sets that are evaluated during simulation run.
Pull	A requisition type that has an ICP requesting supplies from another ICP or local assembly/refurbish resource based on its local inventory position. Pull represents an occurrence of demand somewhere in the supply chain network.
Push	A requisition type that has an ICP sending supplies to other ICPs or a local assembly/refurbish resource based on its local inventory position. Push represents a standing policy to push inventory downstream under user defined conditions.
Adhoc	A requisition type that dynamically determines the replenishment source and quantity during runtime. An internal adhoc requisition uses the predefined list of adhoc types shown in the pipelines block dialog. An external adhoc requisition releases a token into custom modeling constructs to determine the replenishment source, priority and/or quantity. Adhoc requisitioning allows the user to implement smart decision rules that are evaluated as the simulation runs.
Tokens	A custom BALANCE architecture used to provide a means to incorporate new and unanticipated policies into the supply chain simulation, such as for ordering or sending supplies. Tokens are data artifacts generated for all transactions.

2.2 BALANCE Library Architecture

The BALANCE library shown in Figure 2 consists of eleven objects, each containing code for performing that object's intended function. In ExtendSim objects are called blocks and collections of objects may be encapsulated within higher level hierarchical blocks called HBlocks. Each of the individual blocks includes a detailed user help and features guide in the same manner as standard ExtendSim blocks.

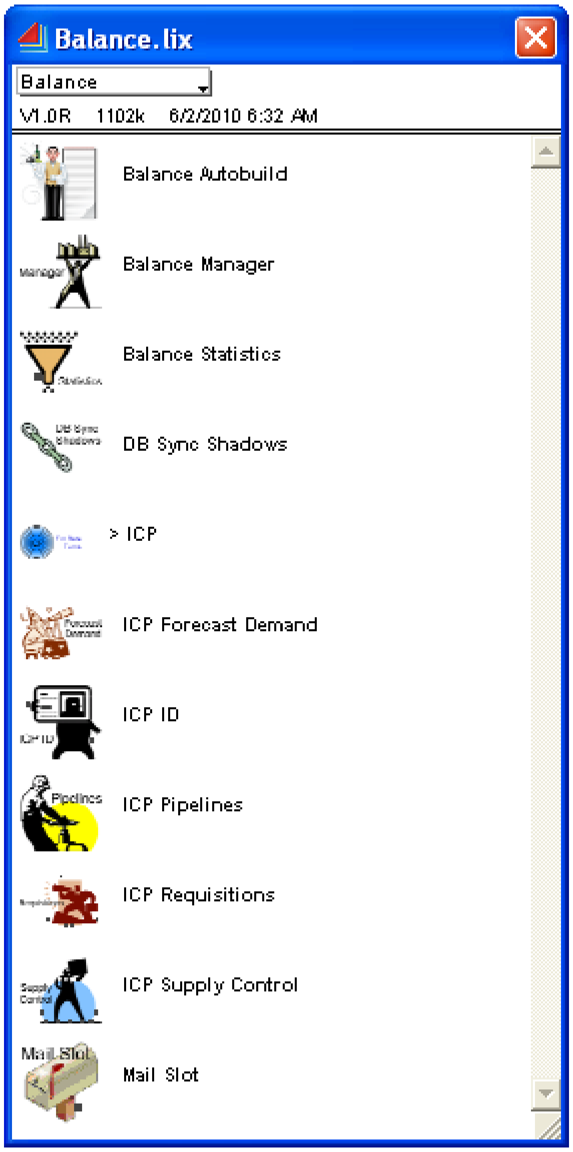
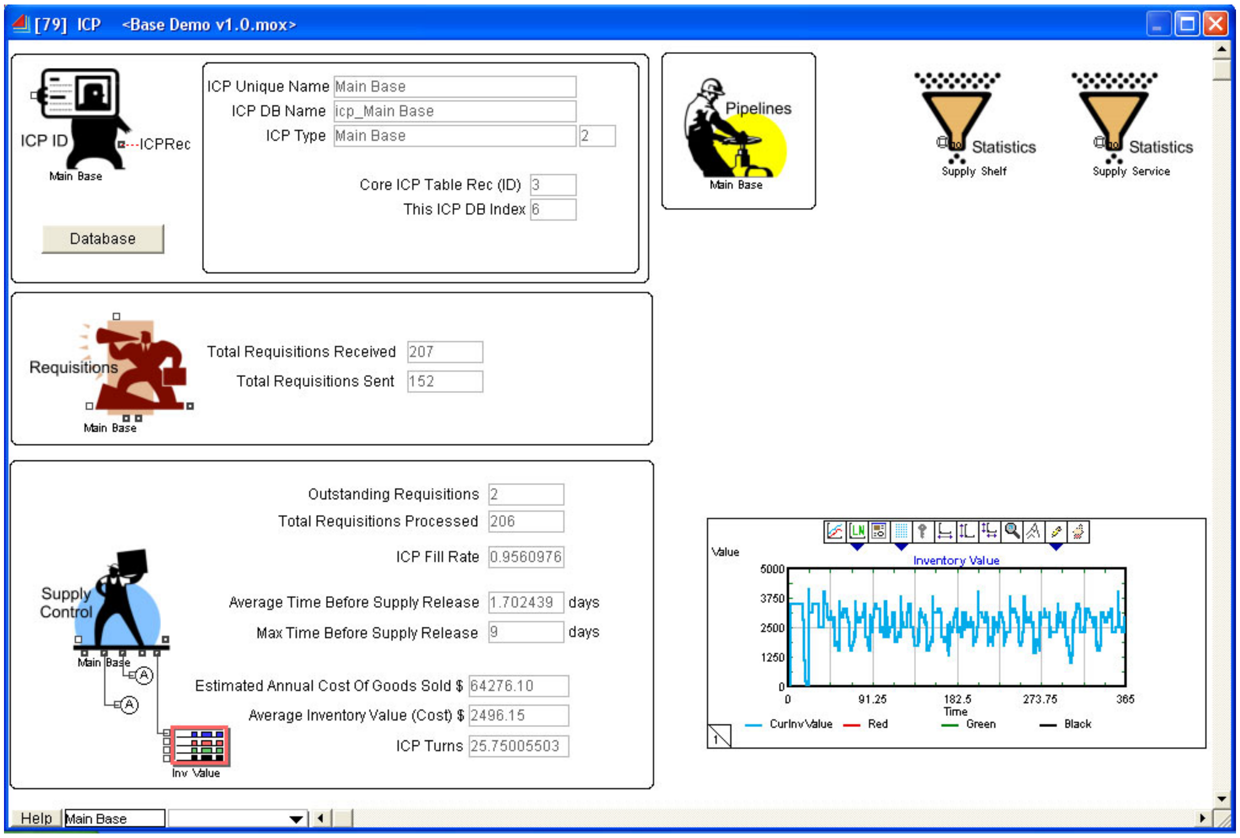


Figure 2: BALANCE library

The principal object in the BALANCE library is the ICP HBlock containing other blocks from the library that perform database management functions, requisition and supply control functions, forecasting functions, and statistics summarization functions. The ICP HBlock is shown in figure 3.



ICP Unique Name: Main Base
ICP DB Name: icp_Main Base
ICP Type: Main Base | 2
Core ICP Table Rec (ID): 3
This ICP DB Index: 6

Total Requisitions Received: 207
Total Requisitions Sent: 152

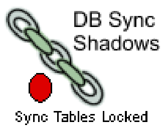
Outstanding Requisitions: 2
Total Requisitions Processed: 206
ICP Fill Rate: 0.9560976
Average Time Before Supply Release: 1.702439 days
Max Time Before Supply Release: 9 days
Estimated Annual Cost Of Goods Sold $: 64276.10
Average Inventory Value (Cost) $: 2496.15
ICP Turns: 25.75005503

Figure 3: Inventory Control Point HBlock

2.2.1 BALANCE Library Objects

BALANCE Manager

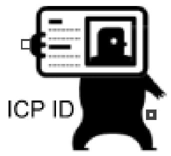
BALANCE models have a single *BALANCE Manager* block placed at the top level. This is the first block placed into a BALANCE model: no other blocks from the BALANCE library can be placed in a model without first having this block in the model. When the block is added, it guides the user through a setup macro that includes the import of the Core BALANCE database, creates and adds a results database and prepares the model for the addition of ICPs.



Database Sync Shadows

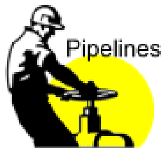
The BALANCE database architecture as described in Section 2.3 contains multiple databases and multiple tables within a database. The *DB Sync Shadows* block maintains common (parent) type and list tables between databases by identifying one database as the “sync” database. In BALANCE models the sync database is the Core database. The block synchronizes all common fields between a sync database table and identical shadow tables in all other databases. The block also monitors and locks the first field of the sync and shadow tables from modification, plus prohibits adding records.

Each BALANCE ICP HBlock contains one of the following four blocks:



ICP ID

Similar to the *BALANCE Manager* block, the *ICP ID* block needs to be the first block placed into an ICP. When added it sets off a setup macro that creates an ICP database. This block provides other ICP blocks with identification parameters needed during configuration and runtime.



ICP Pipelines

The *ICP Pipelines* block provides the dialog used to configure and modify pipelines. This block also maintains the database tables and records to store the information about each pipeline.

ICP Requisitions

The *ICP Requisitions* block is a key runtime block. It takes the timing, rules, and position calculation configured in each pipeline and releases replenishment requisitions based on the current inventory levels and demand. These transactions are recorded by this block in the local ICP database.

Supply Control

The *ICP Supply Control* block is a key runtime block. It takes the requisitions that have been received by the ICP and releases available supplies (shipments). It also receives supplies that are arriving at the ICP. The supply release and deliveries are recorded by this block in the local ICP database.

The following blocks are optional but commonly used in BALANCE models:

ICP Forecast Demand

The *ICP Forecast Demand* block is designed to support models that are driven by forecast tables, such as is typically done when modeling manufacturing process flows to assess overall schedule risk. It can function as a simple top-level demand block during runtime, plus support multi-tiered forecast models by automating the task of building lower tier forecast tables based on pipeline definitions. The block is used in models where forecast tables option was selected in the BALANCE *Manager* block. When this block is added to the model, the *Forecast Demand* database is automatically created and added to the model.

BALANCE Statistics

The *BALANCE Statistics* block accumulates multi-run statistics and confidence intervals. User settings are included to specify whether results will be summarized at the total supply chain network level, individual ICP level, or individual supply level. If **Pipelines** is selected, the block will report results for all pipelines defined in the model. **If Supply Control - ICP** is selected, the block will report results for each ICP Supply Control block. If **Supply Shelf** is selected, the block will report results for on-shelf activity for a single ICP. If **Supply Service** is selected, the block reports results for service performance for a single ICP.

Mail Slot

The *Mail Slot* block was developed to leverage the database link alert capability that is provided with ExtendSim V7. This is useful for setting up models that require some custom ordering or replenishment calculations to be performed based on updates to database cell, field, or table entries. This provides a way using standard ExtendSim functionality to initiate supply requisitions, or other supply chain functions as a consequence of an update or changed database value. Blocks that are monitoring the cell with link alerts will receive a message that the cell has changed. This alert message is then (typically) used to kick off custom model constructs and calculations. This option is valuable in models that require multiple external constructs and calculations and the messaging needs to be event-driven.

BALANCE Auto-build

The *BALANCE Auto-build* block is designed to automatically construct and configure a BALANCE model for the user once the required input tables have been imported describing the network hierarchy, the list of supplies, supply indenture, supply location, reorder and shipment policies and the like. The block then automatically adds ICP modules onto the face of the model, and configures the pipelines, and supply control blocks within each ICP, and sets model run time. This functionality is highly beneficial when building up numerous alternate configured models to support trade studies, or when importing data from logistics data management systems where the end user simply wants to import data using an Extract, Transform, and Load (ETL) tool, push a button and have the model automatically created, configured and ready to run.

2.3 BALANCE Database Architecture

The standardized database that is utilized by the BALANCE library blocks is actually many databases, leveraging the new multi-database capabilities of ExtendSim v7 models. Each database has a specific purpose, dividing tables and data management into natural functionality.

One of the key strengths of the BALANCE library is that a developer never has to build or maintain data structures to support the supply chain model. By adding and configuring blocks in the model, all database constructs are automatically added to the model database via custom developed macros. Compared to other supply chain analysis tools, this automation and data standardization can cut hundreds of hours of initial development time from a modeling project.

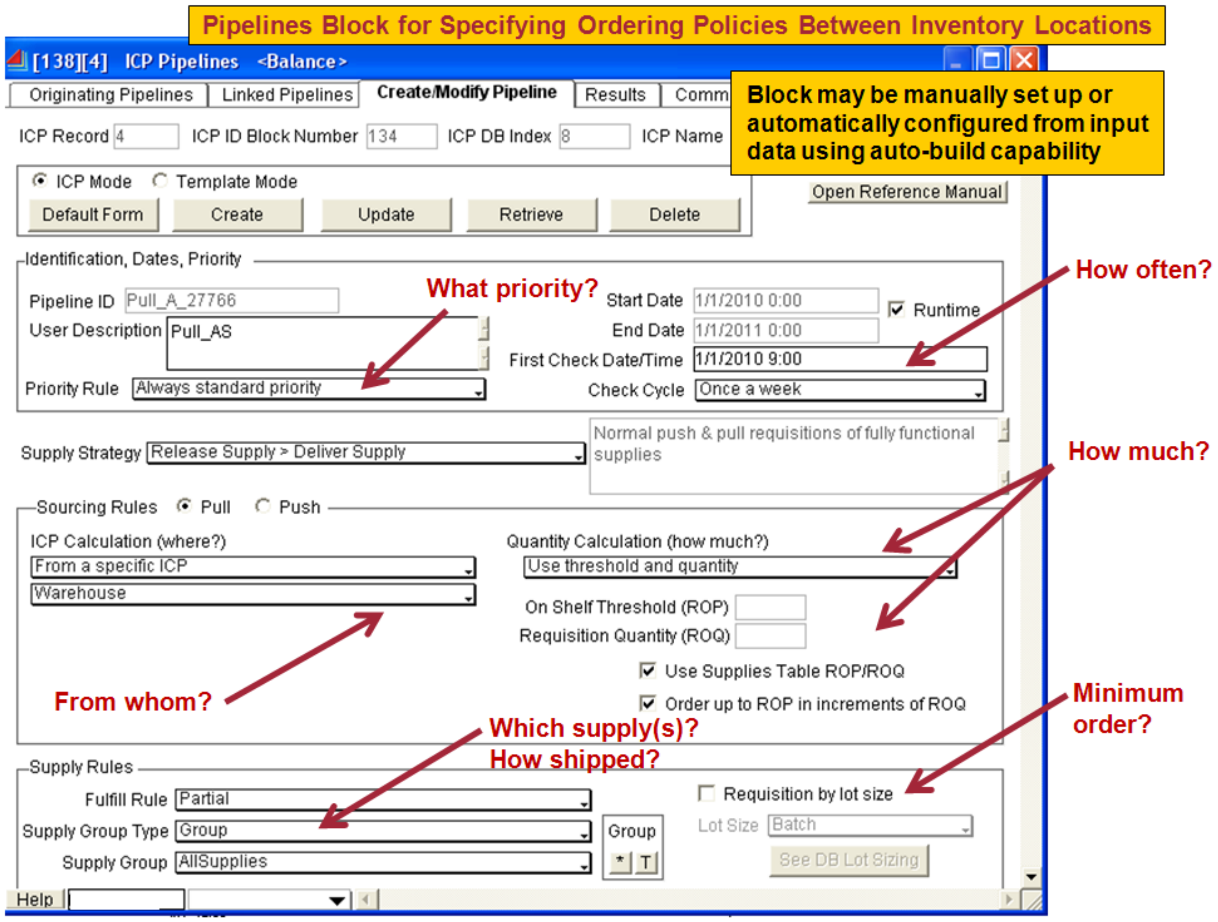
Here is breakdown of the individual database structures in BALANCE and the requirement that is fulfilled by each:

- Core: The database that is automatically constructed when initializing a model with the *BALANCE Manager* block. It is the "highest" database in the framework and has within it the data structure definitions and parent lists for all of the other databases. It also has data structures for detailed ID, status and runtime information concerning each ICP.
- ICP: The database that is automatically added to the model for each inventory control point. Each ICP database keeps detailed information for one ICP, such as pipeline definitions, sent and received requisitions, current inventory levels, part refurbishment information, supply releases (shipments) and runtime results. If an ICP is deleted from a model, the corresponding ICP database is also deleted. The definition detail for this database is stored in two tables in the Core database.
- Results: This database is the collection of runtime ICP transaction records and summary results designed to be exported for post processing. This database is added with the *BALANCE Manager* block during model initialization, and the data is refreshed at the end of each model run. The definition detail for this database is stored in two tables in the Core database.
- Requisition Work: This database is an exposed runtime data structure used by *Requisition* blocks for determining how much supply is needed and which ICP to requisition from. The database is added with the *BALANCE Manager* block during model initialization, and individual tables are added for each pipeline that is created in the model. The definition detail for this database is stored in a table in the Core database.
- Forecast (optional): This database is added when a simulation scenario is utilizing forecasts and the *ICP Forecast Demand* block is added to the model. It holds an exposed data structure that is used when populating the ICP forecast tables. The definition detail for this database is stored in a table in the Core database.

2.4 Graphical User Interfaces for Scenario and Runtime Configuration

The BALANCE library leverages easily understandable user configured block dialogs for model setup. A thorough description of all the block dialog parameters and proper usage is documented in individual block help files and a comprehensive user reference manual, accessible by clicking the *Help* button and *Open Reference Manual* button located on the block dialog. Figure 4 shows the user dialog interface for the *ICP Pipelines* block for configuring ordering polices during model setup and for issuing requisitions as necessary during model runtime in accordance with the policies. Each of the other BALANCE blocks contains a similar user dialog for configuration and runtime execution control.

Drop down menu bars are provided which allow the user to specify pre-coded ordering and shipment policies and rule sets for determining during runtime how often to check if orders need to be placed, who to order from under which conditions, how much to order, which reorder strategy to employ (maintain target inventory or reorder point/reorder quantity, etc.), and shipment related preferences such as allow shipments of partial orders or require complete orders to be fulfilled in a single shipment, plus standard or expedited shipment preferences. In addition to the pre-coded selections, an option called *Adhoc external: release token for calc* option under the *Quantity Calculation* dropdown menu bar provides a mechanism for linking to user–provided custom logic not otherwise covered under the standard provided selections thereby enabling highly tailored logic to be easily implemented as needed.



ICP Record 4 ICP ID Block Number 134 ICP DB Index 8 ICP Name

[✓ ICP Mode] Template Mode

Pipeline ID Pull_A_27766
User Description Pull_AS
Priority Rule Always standard priority
Start Date 1/1/2010 0:00
End Date 1/1/2011 0:00
First Check Date/Time 1/1/2010 9:00
Check Cycle Once a week
[✓ Runtime]

Supply Strategy Release Supply > Deliver Supply
Normal push & pull requisitions of fully functional supplies

Sourcing Rules [✓ Pull] Push
ICP Calculation (where?) From a specific ICP
Warehouse
Quantity Calculation (how much?) Use threshold and quantity
On Shelf Threshold (ROP)
Requisition Quantity (ROQ)
[✓ Use Supplies Table ROP/ROQ]
[✓ Order up to ROP in increments of ROQ]

Fulfill Rule Partial
Supply Group Type Group
Supply Group AllSupplies
Requisition by lot size
Lot Size Batch

Figure 4: ICP Pipelines Block Dialog

3 METRICS AND OUTPUT

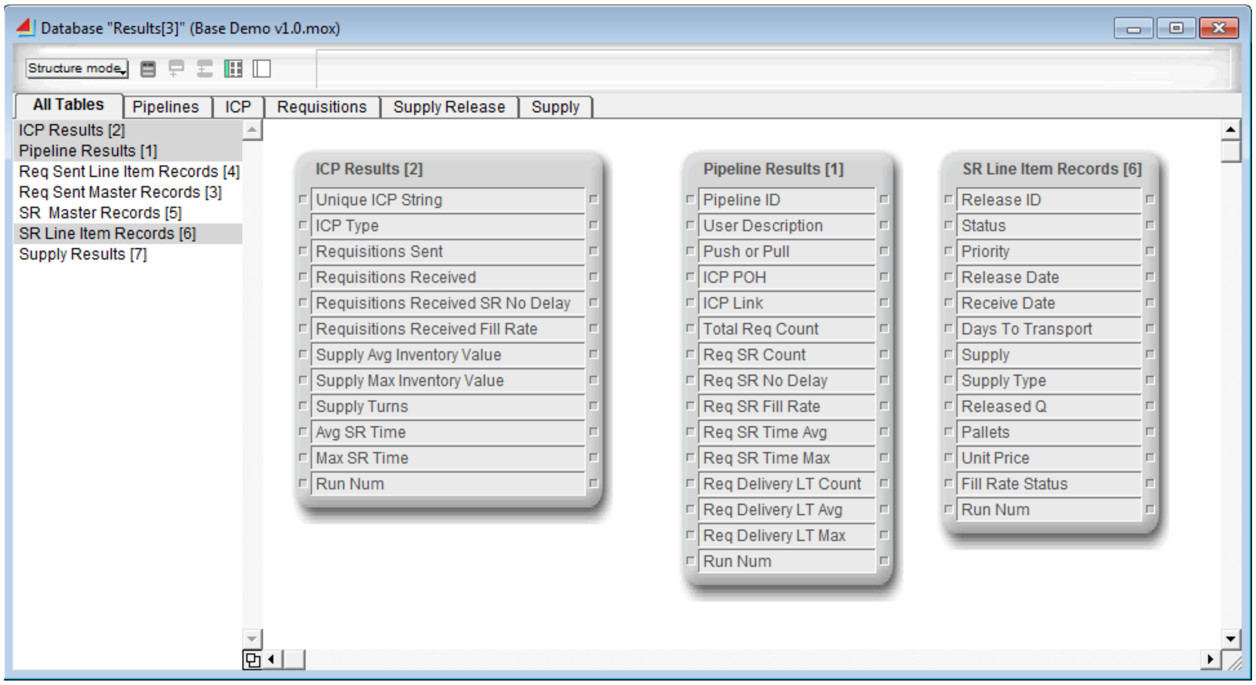
The BALANCE architecture is a discrete-event transaction-based simulation. A large volume of data is captured over the course of a simulation run recording all pertinent details regarding requisitions sent and received, shipments sent and received, current supply positions, inventory levels, inventory values, cash flow and holding costs. This data is captured at the individual ICP level, and summarized across the entire supply chain network.

The principal BALANCE output performance metric is fill rate. Fill rate is defined as the supply quantity shipped without delay divided by the total supply quantity requisitioned. Fill rate is determined when an ICP first attempts to fill (ship) a requisition: if the request can be completely fulfilled, the transaction is logged as 100 percent successful. If the request cannot be completely fulfilled credit is given for the fraction of the requisition quantity that can immediately be fulfilled and the requisition as a whole is logged as delayed. In BALANCE we allow user specified shipment policies such as allowance of partial order shipments and constraints to enforce entire order shipments as desired. These policies can themselves influence supplier fill rate performance.

Fill rate is calculated at three levels in BALANCE: at the ICP level, at the pipeline level, and for individual supplies at each ICP (both inbound and outbound). A user can view the fill rates at each of these model levels to quickly drill down to identify sources of supply bottlenecks and policies that are impacting fill rates.

Inventory turn data is also gathered and summarized by ICP and individual ICP supplies. This key inventory efficiency metric is calculated from annualized cost data and average inventory levels.

BALANCE summary output is captured two ways: a top-level summary Results database is populated at the end of a model run with all requisition and supply release transactions plus inventory turn and fill rate calculations. This single source of output data is designed to be exported to Access or Excel for post-processing of simulation results. Figure 5 shows a depiction of the Results database highlighting three of the key output tables.



ICP Results [2]
Pipeline Results [1]
Req Sent Line Item Records [4]
Req Sent Master Records [3]
SR Master Records [5]
SR Line Item Records [6]
Supply Results [7]

ICP Results [2]
Unique ICP String
ICP Type
Requisitions Sent
Requisitions Received
Requisitions Received SR No Delay
Requisitions Received Fill Rate
Supply Avg Inventory Value
Supply Max Inventory Value
Supply Turns
Avg SR Time
Max SR Time
Run Num

Pipeline Results [1]
Pipeline ID
User Description
Push or Pull
ICP POH
ICP Link
Total Req Count
Req SR Count
Req SR No Delay
Req SR Fill Rate
Req SR Time Avg
Req SR Time Max
Req Delivery LT Count
Req Delivery LT Avg
Req Delivery LT Max
Run Num

SR Line Item Records [6]
Release ID
Status
Priority
Release Date
Receive Date
Days To Transport
Supply
Supply Type
Released Q
Pallets
Unit Price
Fill Rate Status
Run Num

Figure 5: Supply Chain Network-Level Results Database

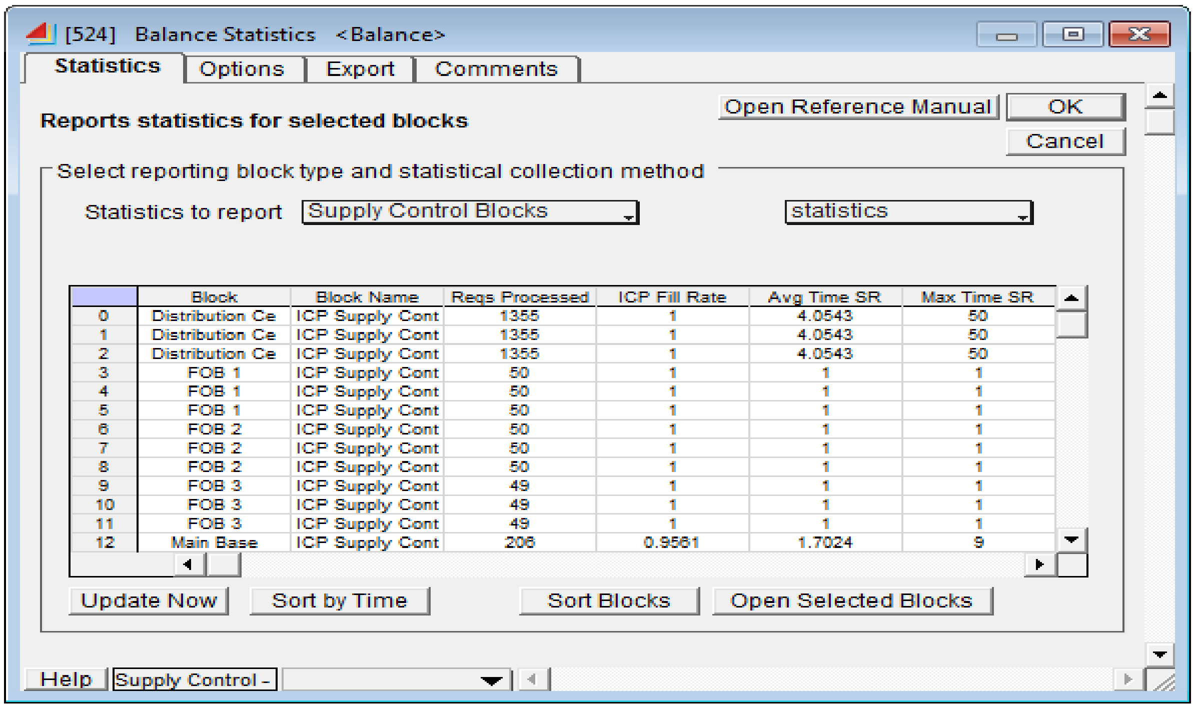
Similarly, statistic blocks are added to a BALANCE model to gather multi-run data for each of the metric levels with quick calculation of mean and confidence intervals. Figure 6 shows a BALANCE statistics block configured to report supply control related information (fill rate at the ICP level).

The summarized output at the ICP level is useful for identifying key suppliers whose fill rate performance is less than desired. By then querying the statistics output for that individual ICP the analyst gains insight into which supply, set of supplies, or particular pipelines are the source of the problem.

Figure 7 shows an example output by individual supply for one ICP.

4 CONCLUSION

The BALANCE logistics analysis capabilities environment simulation architecture has been developed to provide a standardized, reusable supply chain modeling toolset to support inventory management assessment, performance-based logistics contracts risk assessment, and manufacturing schedule risk assessment. A key advantage of the current approach is its integration within a widely commercially available database-driven simulation package. It meets a need for a supply chain and logistics modeling capability that can be applied across a wide spectrum of domains, provides detailed output at the individual supply level, can interface to other logistics and performance analysis tools, can accommodate insertion of custom tailored sub-models for modeling detailed system operations and demand occurrence, and is based on standard commercially available software.



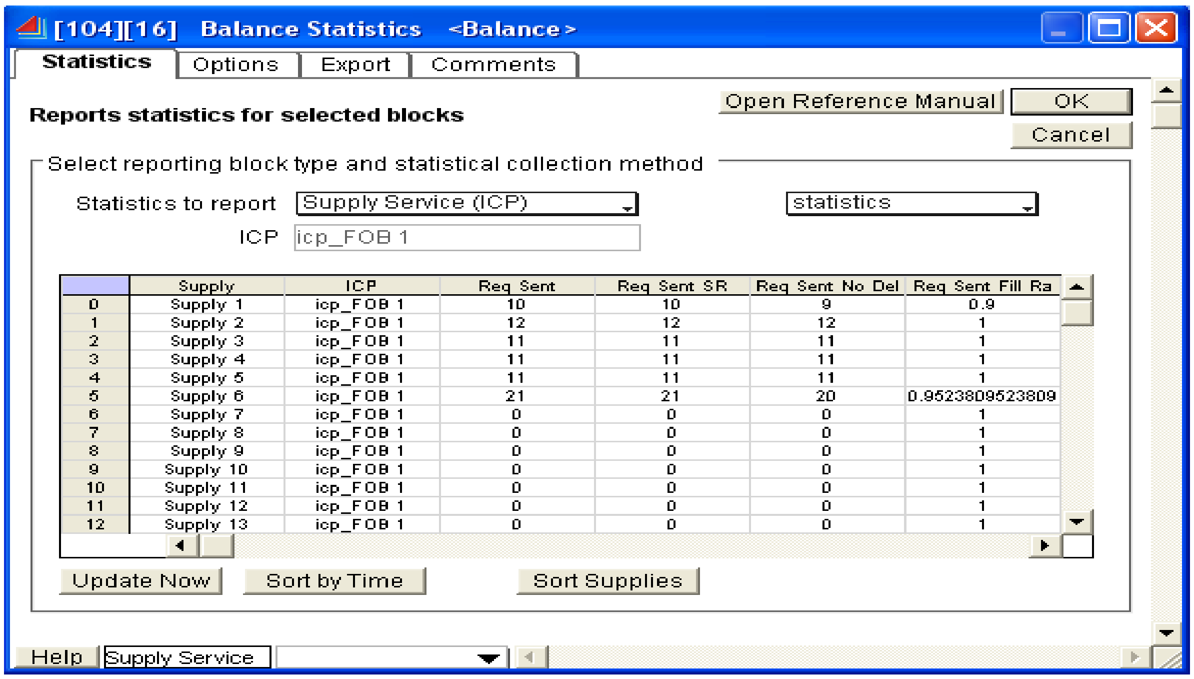
Reports statistics for selected blocks

Select reporting block type and statistical collection method

Statistics to report: Supply Control Blocks; statistics

	Block	Block Name	Reqs Processed	ICP Fill Rate	Avg Time SR	Max Time SR
0	Distribution Ce	ICP Supply Cont	1355	1	4.0543	50
1	Distribution Ce	ICP Supply Cont	1355	1	4.0543	50
2	Distribution Ce	ICP Supply Cont	1355	1	4.0543	50
3	FOB 1	ICP Supply Cont	50	1	1	1
4	FOB 1	ICP Supply Cont	50	1	1	1
5	FOB 1	ICP Supply Cont	50	1	1	1
6	FOB 2	ICP Supply Cont	50	1	1	1
7	FOB 2	ICP Supply Cont	50	1	1	1
8	FOB 2	ICP Supply Cont	50	1	1	1
9	FOB 3	ICP Supply Cont	49	1	1	1
10	FOB 3	ICP Supply Cont	49	1	1	1
11	FOB 3	ICP Supply Cont	49	1	1	1
12	Main Base	ICP Supply Cont	206	0.9561	1.7024	9

Figure 6: BALANCE Multi-Run Statistics Block: ICP-Level



Reports statistics for selected blocks

Select reporting block type and statistical collection method

Statistics to report: Supply Service (ICP); statistics

ICP: icp_FOB 1

	Supply	ICP	Req Sent	Req Sent SR	Req Sent No Del	Req Sent Fill Ra
0	Supply 1	icp_FOB 1	10	10	9	0.9
1	Supply 2	icp_FOB 1	12	12	12	1
2	Supply 3	icp_FOB 1	11	11	11	1
3	Supply 4	icp_FOB 1	11	11	11	1
4	Supply 5	icp_FOB 1	11	11	11	1
5	Supply 6	icp_FOB 1	21	21	20	0.9523809523809
6	Supply 7	icp_FOB 1	0	0	0	1
7	Supply 8	icp_FOB 1	0	0	0	1
8	Supply 9	icp_FOB 1	0	0	0	1
9	Supply 10	icp_FOB 1	0	0	0	1
10	Supply 11	icp_FOB 1	0	0	0	1
11	Supply 12	icp_FOB 1	0	0	0	1
12	Supply 13	icp_FOB 1	0	0	0	1

Figure 7: BALANCE Multi-Run Statistics Block: Supply-Level

REFERENCES

Krahl, D. 2008. ExtendSim 7. In *Proceedings of the 2008 Winter Simulation Conference*, ed. S. J. Mason, R.R. Hill, L. Mönch,, O. Rose, T. Jefferson, and J.W. Fowler, 215–221. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Krahl, D., and S. J. Lamperti. 1997. A Message-Based Discrete-Event Simulation Architecture. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 1361–1367. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Krahl, D. 2002. The Extend Simulation Environment. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J.M. Charnes, 205–213. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Phelps, R. A., D. J. Parsons, and A. J. Siprelle. 2002. Non-Item Based Discrete-Event Simulation Tools. In *Proceedings of the 2002 Winter Simulation Conference*, ed E. Yücesan, C.-H. Chen, J. L. Snowdon, and J.M. Charnes, 182–186. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

U.S. General Accounting Office 2000. *Air Force Operating and Support Cost Reductions Need Higher Priority*. Report to the Chairman and Ranking Minority Member, Subcommittee on Readiness and Management Support, Committee on Armed Services, U.S. Senate. GAO/NSIAD-00-165.

AUTHOR BIOGRAPHIES

STEVEN SAYLOR is an Associate Technical Fellow at The Boeing Company within Boeing Research and Technology (BR&T). His area of focus is in the development of modeling and simulation capabilities for use across the Boeing enterprise. He holds a B.S. degree in Aerospace Engineering from San Diego State University. His email address is <steven.e.saylor@boeing.com>.

JAMES DAILEY is an independent (freelance) simulation analyst based in St. Paul, MN. He has an MBA from the University of Iowa and is Certified in Production and Inventory Management (CPIM) from The Association For Operations Management (APICS). James Dailey is a Consulting Provider for ExtendSim simulation software. His email address is <jim@james-dailey.com>.