

EASY-SC: A SUPPLY CHAIN SIMULATION TOOL

Juqi Liu
Wei Wang
Yueting Chai
Yi Liu

Department of Automation
Tsinghua University
Beijing, 100084, P.R.CHINA

ABSTRACT

To assist the decision makers, we develop a new supply chain simulation software: Easy-SC, a Java-based tool that simplifies the supply chain simulation. In its current state of development, Easy-SC is a modeling tool for assessing the pros and cons of new facility locations, resource allocations and different combinations of policies. It can be used in the modeling of small projects such as single inventory units to large-scale projects such as world wide supply chains. This paper introduces Easy-SC by an examination of its current software module architecture, modeling elements, basic features and simulation processes.

1 INTRODUCTION

Simulation is a powerful methodology which enables decision makers to capture the dynamic nature of complex systems (Chai and Liu 2001). The existing simulation tools can be classified into four categories: (1) integrated simulation environments (2) simulation languages (3) software architectures (4) general-purpose programming languages.

1.1 Integrated Simulation Environments (ISE)

Integrated simulation environments, such as Supply Chain Guru (Technical Assist Company), Supply Chain Builder (Simulation Dynamics Company) and CSCAT (HP/Compaq Company), usually provide friendly graphic user interfaces and support the drag and drop fashion. Users can easily configure one enterprise node through its properties and the simulation processes are fully under control. Moreover, 3D and 2D animations are incorporated within the library objects and are directly linked to the simulation runs. This allows rapid validation and verification of a model. At the end of simulation, vast reports and statistics are presented to assist the analysis work. According to their application areas, these softwares can be of general type (such as Arena) or special type (such as e-SCOR)

1.2 Simulation Languages

Simulation languages are specially developed to describe the simulation problems. Compared with general-purpose programming languages, they are abstracted to a higher level and thus lack the flexibility. Programmers can use them to directly describe the logic relation and topology of one simulation model. Different from ISE, they do not have visualized interfaces. However, the modeling ideas are basically all the same. For example, Arena, one of the most famous simulation software, is based on the SIMAN language (Altiok and Melamed 2001). All the simulation models built in Arena are automatically converted to SIMAN codes before running.

1.3 Software Architectures

One distinguishing feature of software architectures is that they can not be executed, for they are only frameworks where users need to fill the details through further programming. Software architectures, such as Swarm, are usually fulfilled with object-oriented technology and provide class libraries which include:

- Framework: simulation engine, event table, base classes and interfaces.
- Utility: random number generator, math functions, etc.

In order to build a simulation model, users need to extend the base classes, overload the virtual methods and implement some callback functions which will be called during the simulation.

1.4 General-Purpose Programming Languages (GPPL)

Actually, all the simulation softwares are implemented with general-purpose programming languages. They have the maximum flexibility but are not capable to model complex

systems. Therefore when the problems are small enough to be dealt with hundreds lines of codes or they are too specific to find any suitable softwares, people will try to use general-purpose programming languages to build the simulation models. Such languages include C/C++, Python, Java, etc.

The basic features of the four kinds of simulation tools are summarized in Table 1.

Table 1: Basic Features of Different Simulation Tools

Feature	ISE	Simulation Languages	Simulation Frameworks	GPPL
Visualized modeling	Yes	No	No	No
Animations	Yes	No	No	No
Programming	No	Yes	Yes	Yes
Flexibility	Lowest	Lower	Higher	Highest

Although many tools can be used in supply chain simulation, there are still some limitations:

1. Many ISE, such as Arena, Simprocess and Log-Sim, use abstract modules to build simulation models. While this method can greatly enhance the flexibility, it will increase the difficulty in mapping between reality and simulation. Users have to decompose the real enterprises, such as suppliers and distribution centers, into basic logic modules and it is hard for others to grasp the gist of the model at the first glance.
2. Simulation Frameworks and general programming languages do not provide friendly modeling environments, and they require users to do some programming work before simulation. Professional skills are needed in these processes and it makes these two tools not quite fit for the beginners

2 OVERVIEW OF EASY-SC

We try to design a new supply chain simulation tool which integrates the advantages of both ISE and software architectures. It is implemented as an open extensible framework and can simplify the modeling process through a graphic user interface.

2.1 System Goal

Easy-SC is a simulation platform for understanding supply chains through studying the impact of stochastic demands, logistics decisions and production policies on key performance measures. It is called “easy” for the following four reasons:

2.1.1 Easy Modeling

Easy-SC allows users to graphically build a supply chain through a graphic user interface and then specify relevant logical and descriptive information to fully define the supply

chain. Also included in a description of supply chain logic are policies that define conditions under which actions occur by describing circumstances that “trigger” the actions or by defining goals that are to be met because of performing the actions. Examples include inventory, transportation and production. Moreover, 2D animations are shown during the simulation processes and results are organized into various forms such as charts, tables and plain texts.

2.1.2 Easy Extension

Although our main goal is to provide a specialized simulation environment, we try to design the architecture as general as possible. Users can add their own business processes into the system.

2.1.3 Easy Description

There are at most six units in one supply chain enterprise node, and each unit has its unique “inactive” state in which it performs nothing but delivers the information to the next unit. Through this function, users can focus on the local performance without specifying all the descriptive information. It turns out to be especially useful for departmental managers.

2.1.4 Easy Integration

Easy-SC is written in Java language so that it is platform independent and can be integrated into management information systems which are based on J2EE seamlessly.

2.2 Main Structure

The main structure of Easy-SC involves the following three components, each of which will perform an important role in the system:

2.2.1 Module Architecture

In Easy-SC, modules are the basic simulation units. They consist of nodes (or enterprise entities within the supply chain), and arcs (or pathways that connect the nodes). General topology of the supply chain is defined by the sets of nodes and arcs.

There are two kinds of variables in one module: state variables and property variables. State variables reflect the changes of module states and keep varying during the simulation. We can evaluate the module performance and generate the final reports based on these variables. Property variables are detailed parameters entered by users. They represent the module’s inner characteristics, control its activity and do not change during the simulation. Such variables include safety inventory, reorder points, product prices and so on.

We also provide some module tools such as:

- **Module manager.** Module manager is in charge of creating, managing, and destroying modules. It maintains a module list which is used to get and set the module information. Moreover, we can easily get one module's handle from the list by its ID number.
- **Simulation engine.** Simulation engine determines the way how the simulation clock goes (Banks et al. 2001). There are two working modes for the simulation engine. In the uniform mode, the simulation clock is synchronized to the system clock. Step by step, system checks the event list and trigs the event that should happen at the current time. In the jumping mode, the simulation clock jumps to the time when the next event should happen. Simulation runs much faster in the jumping mode, however, animations can only be shown in the uniform mode.
- **Mail.** Mail is used in the communication between two modules. It is scheduled by the simulation engine which determines its sending time and transmission delays. Each mail carries a task that will be executed as soon as it is received.
- **Blackboard.** Blackboard is a supplement to mail. Whenever a module wants to share some information, it posts "notices" on blackboard so that other modules who are interested in this kind of information can check their content. Blackboard facilitates the request-response process and can be used to test the effect of information sharing.

The relationships of the four tools are illustrated in Figure 1:

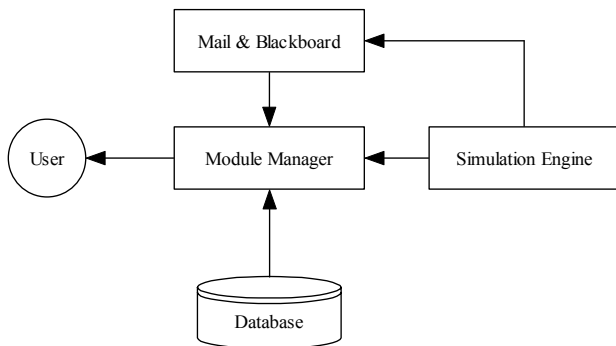


Figure 1: Relationships of the Four Module Tools

2.2.2 Utility

Utility is a set of tools to help users build the simulation models. It includes:

Debug tools: assist users to find and solve the simulation problems

Graphic tools: modify the drawings that are automatically generated by Easy-SC.

Locale Tools: switch the show language between Chinese and English.

2.2.3 User Interface

Graphic user interface can visualize the modeling processes, show animations and make the simulation results more understandable. Java Swing library and JFreeChart are used in the interface programming. A screen snapshot is shown in Figure 2.

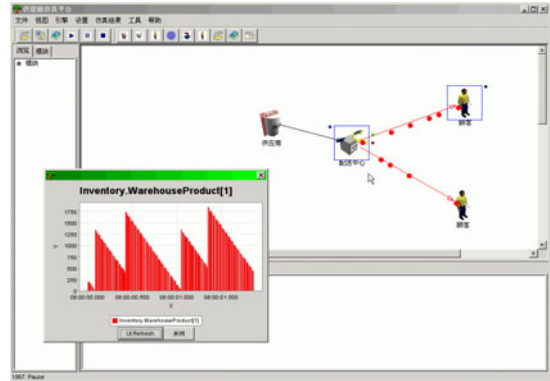


Figure 2: A Simple Supply Chain Model

3 SUPPLY CHAIN MODEL

The basic supply chain layout is built in a drag and drop fashion, resulting in a set of nodes and connection arcs that define the general topology of the supply chain. Moreover, users can further describe the transportation network with connection arcs and transshipment nodes which link connection arcs together.

3.1 Enterprise Nodes

The node types that Easy-SC allows users to create include suppliers, distribution centers, retailers, manufacturers, customers, and carriers. The basic features of these nodes are listed in Table 2.

Table 2: Enterprise Nodes

Name	Features
Suppliers	The most upstream enterprises. Have infinite inventory
Distribution Centers	Distribute products and have no manufacturing ability
Retailers	Have no inventory
Manufacturers	Have the ability to transform raw materials into finished products
Customers	Generate the original downstream demands
Carriers	Provide the third party transportation service

3.2 Processing Units

Processing units provide the basic logistics functions in enterprise nodes. Processing units can be of six types: source, deliver, make, transport, inventory and finance. Which kinds of units does a node contain is determined by the type of the node. (For example, DC nodes do not contain manufacture units). The relationships of the six units are shown in the following figure.

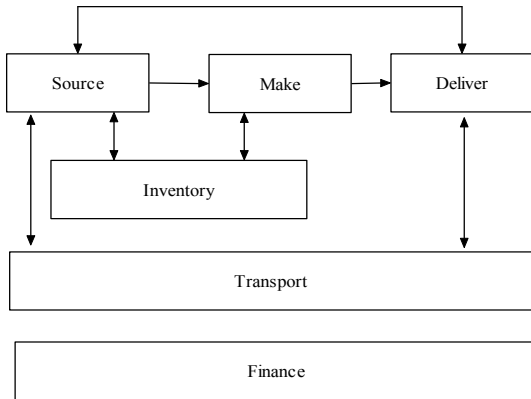


Figure 3: The Relationships of Processing Units

“Source” units are in charge of combining purchasing requirements sent from inventory units, selecting suppliers according to the user defined policy and placing orders.

“Deliver” units generate forecast sequences according to the customer demands. They also manage orders by assigning them priorities, determining the execution time and making combinations to reduce the handling costs.

“Make” units transform raw materials into finished products.

“Transport” units are responsible for the shipments between different nodes. They will find the best transportation route, mode (road, rail or air) and schedule to ship the finished orders.

“Inventory” units control the amount of raw materials and finished products that are stored in the warehouse. They generate purchasing requirements according to the inventory policy, forward them to the source units, and receive the replenishment goods.

“Finance” units are responsible for the cash flows which include payments, money receptions, and various costs.

3.3 Connection Arcs

An arc represents one transportation path between two nodes. Users can add different transportation modes into one arc and specify their relevant descriptive information such as distances, costs, and damage rates. This information can be used in transportation decisions.

3.4 Other Elements

Beyond the basic topology defined by nodes and connection arcs, there are also some global elements in Easy-SC.

3.4.1 Products

We separate the properties of one product into two groups. The first group is about the general properties and the second group is about the specific properties that vary from one node to another. Users can add a new product globally to the system by setting its general properties such as ID, name, volume, and weight. Easy-SC manages these products through the product manager. In addition, each enterprise node maintains a product list in which the specific properties about the price, amount, and inventory control policy of each product contained in the node are recorded.

3.4.2 Resource

Resource is a broad concept in simulation. In Easy-SC, all the resources are classified into three categories: warehouse, transportation and manufacturing. They are responsible for the corresponding delays and costs. Users can add new resources into the three categories and specify their names, delays and unit costs.

4 SIMULATION PROCESS

A node in a supply chain performs five basic actions with regard to the life cycle of an order:

4.1 Sourcing Task Creation

Sourcing task creation involves the creation and initialization of a sourcing task. Sourcing tasks are generally created by one of two sources, an inventory policy signaling that a replenishment is necessary, or a sourcing plan made according to the node’s demand feature. At the time of creation, a task which includes the information about desired products and each product’s quantity and time window is generated. Sourcing tasks are automatically transmitted to the sourcing unit.

4.2 Order Creation

Order creation is a process in sourcing unit. It generates orders according to the sourcing tasks. It may include combining sourcing tasks, selecting the suppliers, allotting the quantity among selected suppliers and sending out orders to the target nodes.

4.3 Order Processing

Order processing is the action that attempts to meet the needs or demands of orders. It is first processed by the “or-

der handling procedure” which is part of the “Deliver” unit. The procedure calculates order priorities and then arranges orders in a queue. System checks the order queue periodically, finds the orders that should be processed and changes their states. Fulfillment of an order is made from the finished products inventory. If finished products inventory cannot meet the order’s need, the order must be divided into several smaller orders or wait until new products arrive in the inventory. Finished products arrive as a result of either a processing delay (for supplier nodes) or a production process (for manufacturing nodes). The completion of the order processing is the point at which the order is sending back to its origin. The delays and costs involved with order processing are dependent on the individual steps such as order handling, outbound process and so on.

4.4 Order Transport

Order transport is the process of preparing the fulfilled or part-fulfilled order for transport to its origination node. Order transport is part of the “Transportation” unit. It may involve grouping of certain orders together, selecting the transportation partner and determining the best route. Costs and delays are involved in the transport process.

4.5 Order Receiving

Order receiving is the process of accepting an order that has been filled. It first generates an inbound task to arouse the inventory unit, then creates a payment task to pay a sum of money to the node which has fulfilled the order. An order’s lifecycle is finished after the payment task is completed.

4.6 Order States

An order’s state keeps varying during the simulation. All the order states are listed in Table 3, and Figure 4 shows their transition relationships.

5 OTHER FEATURES

5.1 Unit Simulation

Input required for the successful design and execution of a supply chain simulation model is extensive and various. But for some departmental managers, it’s difficult and not necessary to obtain the detailed information of all the units. What they care about is only the performance of their own department. So we provide the unit simulation function to help users build simple models where some units are set to their default states and are inactive during the simulation.

In Easy-SC, each unit can be set to “active” or “inactive” state. In the active state, users must input the value of each property and specify all the policies. The active unit then performs normally during the simulation. However,

Table 3: Order States

State	Meaning
Created	Created by customers but is still on holding;
Waiting	Waiting in the order queue
To be handled	There are enough products in warehouse;
Handling	Inbound process
Unsatisfied	There are not enough products. The order can not be handled right now
Abandon	The order can not be finished in time. Remove it from the queue
Transporting	Outbound process
Finished	The finished order has been received by the customer who sent it
Archive	Provide statistics data
Abnormal	Unexpected states

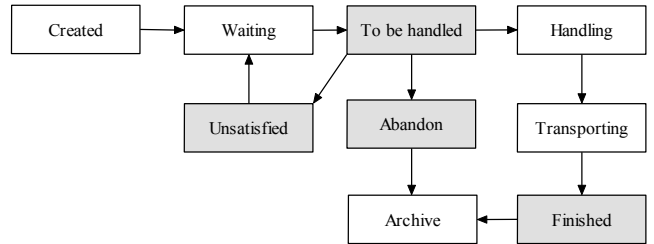


Figure 4: The Transition of Order States

once a unit is set to the inactive state, all the inner properties and policies are automatically set to their default values and the unit performs nothing but forwards the information to the next unit. For example, an “inactive” inventory unit has infinite products and warehouse resources so that there are no out-of-stock possibility and handling delays. It immediately transforms the order handling task to the shipping task and passes it to the transportation unit. If all but the transportation unit in an enterprise node are set to the inactive state and customer nodes are used to send orders representing the arriving process of shipping tasks, we are able to observe the performance of the transportation unit simply and clearly. Similar models can be used to simulate the inventory unit, the source unit and so on.

5.2 Policies and Optimization

One distinguishing feature of Easy-SC is that it provides plenty of choices at each decision point and can suggest the optimized values for some properties. We believe that the main operation processes in all supply chain are basically the same and the difference comes from the policies adopted at each decision point. So we accomplish vast policies that are commonly used. Users can easily select these policies according to their own business process, or change some policies and test the effects of different combinations.

Moreover, we integrate many optimization methods to help users determine the values of some critical properties. For example, in inventory units, various optimization models are used to calculate the reorder points, order quantities,

and order periods in different situations. With their help, users are freed from calculation works and the optimized values can be taken as instructions to real problems.

5.3 Location Tool

In the past years, facility location has been widely adopted as a critical aspect of strategic planning for a broad spectrum of public and private firms. Whether a retail chain sitting a new outlet or a manufacturer choosing where to position a warehouse, strategic planners must identify good potential locations, determine appropriate capacity and allocate large amount of capital. The high costs associated with this process make almost any location project a long-term investment. Determining the best location for new facilities is thus an important strategic challenge. In Easy-SC, we develop the location tool to help users find the ideal locations of new nodes. The optimization result includes the new node's location, capacity and supply relationship with other nodes. Several classic models like p-media, p-center and max covering are integrated in this tool. We also develop some new stochastic models to make the optimization results more reliable.

5.4 Transportation Network

Different from other simulation software, Easy-SC uses not only arcs but also transshipment nodes to model the whole transportation network. An entire route between two enterprise nodes can be divided into several fractions and users can update the information of one fraction without changing the others. Generally speaking, the transportation arcs are used to model the paths which have been adopted in the real world. There can be more than one transportation mode in one arc, each with its unique distance, cost and damage rate. With this information, transportation units are able to find the best node-to-node or circle routes which have the shortest distance, minimum cost or least damage rate for shipping tasks. Moreover, using the optimized route is only one choice; users can also freely specify another route by themselves.

6 CONCLUSIONS

Simulation is an effective way to get the detailed answers and to consult the customers on how a supply chain should be run. As a fully object-oriented supply chain simulator, Easy-SC goes beyond current simulators by enabling GUI-based "off-the-shelf" modeling of common supply chain models while still providing an open framework for customization if desired. The Java-based core of Easy-SC provides a robust, flexible, internet-friendly, and multi-threaded environment for simulation model development and execution. With unit simulation, Easy-SC can handle models of various scales. Moreover, vast policies and op-

timization models are integrated to lower the "barriers to entry" for simulation modeling of supply chains.

REFERENCES

- Altiok, T., and B. Melamed. 2001. *Simulation Modeling and Analysis With Arena*. Cyber Research Inc. and Enterprise Technology Solutions, Inc.
- Banks, J., J.S. Carson II, B.L. Nelson, and D.M. Nicol. 2001. *Discrete-Event System Simulation*, Prentice Hall.
- Chai, Yueting, and Y. Liu. 2001. *Agile supply chain management*. Tsinghua University Press.

AUTHOR BIOGRAPHIES

JUQI LIU is a master candidate of Automation Department at Tsinghua University. He received his bachelor degree from Tsinghua University in 1998. His research interests include facility location, logistics simulation and supply chain management. His email is <ljq02@mails.tsinghua.edu.cn>

WEI WANG is a master candidate of Automation Department at Tsinghua University. He received his bachelor degree from Tsinghua University in 1998. His research interests include logistics simulation and supply chain management.

YUETING CHAI is a professor of Automation Department at Tsinghua University. His research interests include e-business, enterprise collaboration and agile supply chain management.

YI LIU is an associate professor of Automation Department at Tsinghua University. His research interests include e-business, ERP and agile supply chain management.