

## EXTENDSIM 7

David Krahl

Imagine That, Inc.  
6830 Via Del Oro, Suite 230  
San Jose, CA 95119, U.S.A.

### ABSTRACT

The ExtendSim simulation environment provides the tools for all levels of modelers to efficiently create accurate, credible, and usable models. ExtendSim's design facilitates every phase of the simulation project, from creating, validating, and verifying the model, to the construction of a user interface which allows others to analyze the system. Simulation tool developers can use ExtendSim's built-in, compiled language, ModL, to create reusable modeling components. All of this is done within a single, self-contained software program that does not require external interfaces, compilers, or code generators.

### 1 INTRODUCTION

Building on Extend's reputation as a workhorse for large-scale graphically based simulation models, the recently introduced ExtendSim 7 demonstrates Imagine That's commitment to supporting a wide range of modelers. Extend (now ExtendSim) has traditionally been the tool of choice for the most challenging of simulation problems.

The underlying technology of a simulation application can have a significant impact on software features and capabilities. A solid basic design efficiently supports new technologies and customer demands. ExtendSim's architecture continues to lead the industry in extensibility and robustness.

ExtendSim was designed from the ground-up to support large applications and rapid model deployment. Features—such as hierarchical modeling, interactive model building, and fast execution speed—support this. These features, in turn, are made possible by ExtendSim's underlying architecture.

### 2 EXTENDSIM 7

ExtendSim 7 is built on the capabilities of Extend 6 with specific attention to features that enhance presentation, usability, and scalability. These improvements were based on the experiences of Imagine That's large customer base, in-

put from Imagine That's most advanced customers, and genuinely inventive ideas from Imagine That's engineering staff.

The end result of this development is that modelers will require fewer steps to create a more efficient and attractive model. But perhaps the most significant aspect of ExtendSim 7 is that it creates a solid base for future development. In short, ExtendSim 7 is only the beginning of a new generation of software from Imagine That, Inc.

### 3 EXTENDSIM BASICS

Before looking into ExtendSim's technology, it is helpful to understand the ExtendSim modeling environment (Imagine That, Inc. 2003).

ExtendSim models are constructed with library-based iconic blocks. Each block describes a calculation or a step in a process. Block dialogs are the basic mechanism for entering model data and reporting block results. Blocks reside in libraries where each library represents a grouping of blocks with similar characteristics. Blocks are placed on the model worksheet by dragging them from the library window onto the worksheet. The flow is then established between the blocks.

In a typical discrete event model, blocks from the Item and Value libraries will make up most of the model. A typical block from the Value library would be a Lookup Table block:



Figure 1: Lookup Table

Blocks from the Value library perform mathematical calculations and have value connectors for their inputs and outputs. Figure 1 shows a Lookup Table block that performs a lookup converting one value input (□) to any number of outputs (■) based on its internal table. The pull down inverted triangle (▼) indicates that this connector can be expanded.

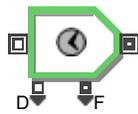


Figure 2: Activity

Blocks from the Item library operate on items (entities). Figure 2 shows an Activity block from the Item library. The Activity has item inputs (□) and item outputs (□) to define the flow of the item through the block and value inputs and outputs for the parameters and results of the activity. Activities have a limited or infinity capacity, delay items for a fixed or random amount of time, and may be shut down, interrupted, or suspended. This particular Activity has only some of its connectors visible. Other connectors become available when the modeler selects the appropriate options in the block or if they pull down on the inverted triangle.

Table 1: ExtendSim’s standard libraries

Library	Description
Item	Item processing blocks
Value	Value processing blocks
Plotter	Plots and charts
Animation 2D 3D	Animation for 2D and 3D environments
Rate	High speed or rate based processes
Utilities	Blocks that perform utility functions
Electronics	Electronic components

Table 1 shows the main libraries available with ExtendSim from Imagine That, Inc. Other example libraries that illustrate modeling and programming techniques are included as well.

### 3.1 Building a model

Building a model involves dragging a block from the library into the model window, connecting it to the other blocks in the model, and then setting the block’s dialog parameters. This connection may be made by using the block’s connectors or it may be a logical connection such as choosing the name of a resource in a queue block.

A variation on the classic bank line model, this example will add balking and renegeing to a basic M/M/1 simulation. Customers arrive to the bank, wait for a teller, are helped by the teller and then leave. However, if the line is too long when they arrive they may randomly decide to balk and leave the bank. If a customer waits too long in line, they will renege and leave the bank.

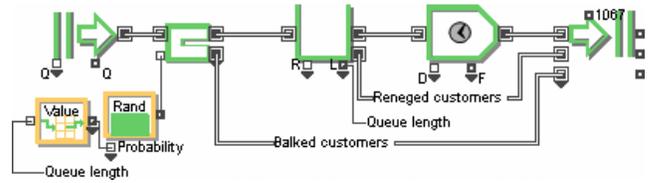


Figure 3: Bank line with renegeing and balking

The model in Figure 3 illustrates a number of features in ExtendSim:

- The use of the connection lines to transfer items (customers) as well as information (probability of balking) through the model.
- The “Queue length” named connection establishes a connection through two identical text objects as if they were physically attached. This helps to organize and document the model.
- The relationship between the length of the queue and the path that the item will take can be clearly seen. It is also obvious that this the path selection is dependant on a random variable. In this case, there is a random probability that a customer will balk based on the length of the queue.
- Paths for the items through the model are clearly visible.
- The number of inputs to the Exit block has been expanded so that all of the items (customers) leaving the model go to the same block.

### 3.2 Hierarchy

Even in this example, it is apparent that a model with a large number of blocks on the same worksheet could be cumbersome to work with. To organize models, ExtendSim includes hierarchical modeling—combining multiple blocks into a single block. Hierarchical blocks can be up to 255 levels deep and can contain any number of other blocks.

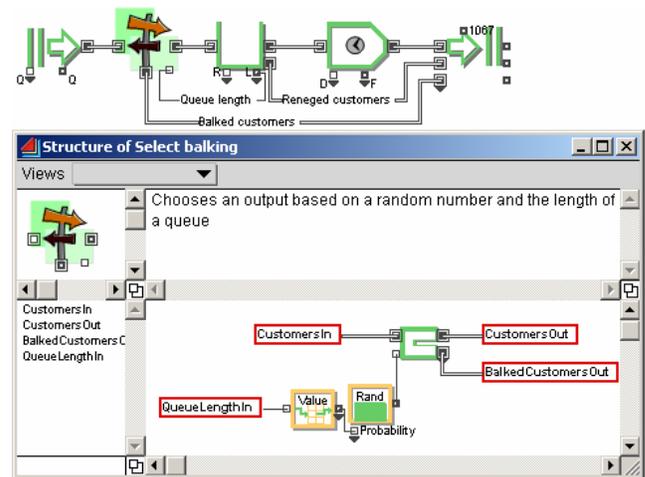


Figure 4: Bank line with hierarchy

In Figure 4 it can be seen how the top level of the model is simplified by combining the three blocks that choose which way an arriving customer will go into a single, hierarchical block. A more descriptive icon and help have been added.

### 3.3 Cloning

When hierarchy is used in a model, some of the important parameters and results tend to get pushed down into the model so that they are not easily accessible. ExtendSim incorporates an innovative feature called cloning that allows the modeler to pull important inputs and outputs to the interior of a hierarchical block, the model worksheet, or the model notebook. Cloning is done by selecting the clone tool (Ⓢ) and dragging any dialog item to a new location.

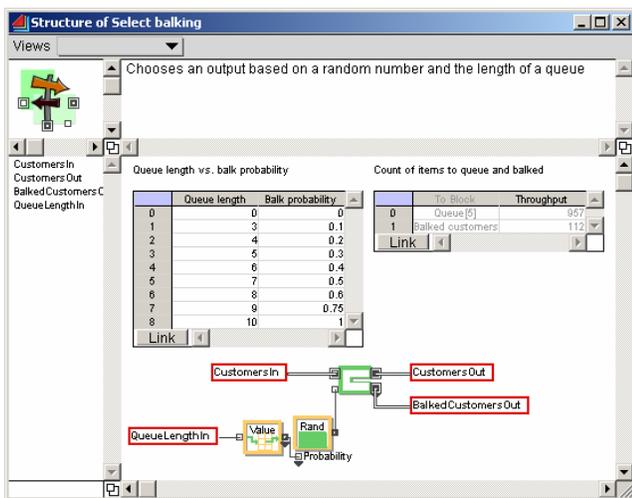


Figure 5: Cloned dialogs added to hierarchical block

Cloned dialog items can be either inputs or outputs. In Figure 5 the lookup table from the Lookup Table block and the results table from the Select Item Output block have been cloned to the hierarchical block's worksheet. In doing this, it is now obvious what the probabilities are for balking and the number of customers that entered the queue and the number that bailed. Note that through either the clones or the original block dialogs, the probabilities can be changed and the results will be updated during the simulation run without any programming on the part of the modeler.

### 3.4 Animation

ExtendSim incorporates animation for model communication, debugging, and presentation. By default, models are animated on the worksheet, providing a quick view of the workings of the simulation. A more sophisticated worksheet animation can be created by animating the hierarchi-

cal blocks and changing the pictures that represent the items. This is all done within the standard ExtendSim drag and drop interface.

A 3D animation can be created by opening the 3D window. This enables the 3D "QuickView" mode. In this mode the item handling blocks in the model will have a representation in the 3D world and items will be animated as they move from block to block.

A more lifelike 3D animation can be created by changing to the concurrent or buffered modes. In these modes a constant ratio between simulation time and real time is maintained. Additional animation features such as mounting items together to combine them, adding scenery to show background objects, hiding selected blocks, and defining paths for the items to follow can be added to improve the look and accuracy of the animation. 3D animation actions are built into ExtendSim's Item Library blocks. These can be augmented by blocks from the Animation 2D-3D library, equations, or custom built blocks.

Notice how the 3D environment is completely integrated with the simulation environment. However, 3D animation can be added at any stage, but is not a required part of the logical model building process.



Figure 6: 3D Animation of Airport Security

## 4 TECHNOLOGY

ExtendSim's underlying technology is the key to its unique combination of flexibility and ease of use. Technological features such as integrated databases, data linking, discrete rate modeling, value and item messaging, and a built-in development environment make ExtendSim stand out among simulation programs.

### 4.1 Integrated Database and Data Linking

Ask any experienced modeler, the most significant challenge to simulation modeling relates to the data—its collection, organization, and maintenance. As computing

power has increased, larger and more comprehensive simulation models are being built. In the 1980's a large model encompassed an entire manufacturing plant. Now, multiple manufacturing operations, warehouses, and consumers are typically modeled as a single supply chain. These increases in model size have correspondingly increased the amount of data that is required for the model. While one farsighted simulation software company (Pritsker and Standridge, 1987) had developed databases integrated with their simulation software in the 1980's, the state of the art at that time in computer performance, user interface, and the data requirements meant that that the data management features were rarely used.

In the 1990's Simulation Dynamics, in conjunction with Imagine That, Inc. developed a second generation simulation database (Imagine That, Inc., 2001) incorporated into the Extend + Industry product. This database has proven invaluable to modelers that need to manage and organize large amounts of data. Simulation applications utilizing the Industry database ranged from military supply chains (Hamber, 1999) to Construction (Angelides, 1998) to processing nuclear material (Kress, 2006).

Imagine That's third generation graphical simulation database (GSDB) incorporates an improved user interface, better integration with the simulation model, and new technology for ensuring that the information is relayed efficiently and can be effectively used to control any detail in the simulation model.

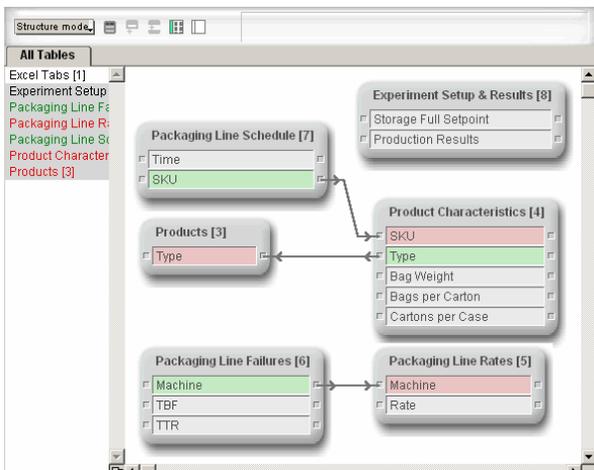


Figure 7: ExtendSim's Graphical Simulation Database

ExtendSim's GSDB is a relational database. This allows the modeler to efficiently manage the data in a simulation model. Figure 7 shows a database for a small packaging line model. The relations between the fields in the data tables are clearly visible.

One of the challenges in merging a database and a simulation is the actual transferring of data. In other simulation tools, middleware must be written in another language, such as C++ or Visual Basic, to communicate be-

tween a the simulation and the database. In ExtendSim we have provided three different mechanisms for communication between the database and the simulation model:

- Dynamic data linking (DDL). Dialog parameters and tables can be linked to a database record or table. This is done by clicking on the "Link" button (see cloned tables in Figure 5) and selecting the appropriate table and/or record. ExtendSim's Link Alert technology automatically notifies the block when a database value has changed. The block can react immediately, potentially changing the status of the simulation model. With this technology, polling the database at the end of each simulation step is not necessary, since changes to the records in the database are reacted to immediately.
- Read and Write blocks. ExtendSim 7 includes blocks in the Item and Value library that can read and write to the database. The database, table, field, and record can be controlled with a connector. This allows the modeler to traverse one or more databases during the simulation run.
- A full set of database functions. There are over 40 new functions that allow the modeler to create, access, and manipulate databases and their contents. These can be called from an equation or by a custom block.

## 4.2 Discrete Rate

Discrete rate simulation uses discrete event clock scheduling to simulate high-speed or rate based systems. Rather than individual items moving from one block to another, discrete rate simulates a flow of material moving through the model. This differs from a continuous model in that the simulation clock moves from one event to the next and is not based on a fixed step size. Events in a discrete rate model include a tank filling up, a rate change due to a product change-over, an equipment breakdown, or an arrival of a discrete event item representing a batch.

To provide global oversight and calculate the effective rates in a discrete rate model, ExtendSim uses linear programming (LP) technology. After all the rules for storage capacity and movement have been declared in the model, ExtendSim uses an LP calculation to maximize the flow through the system. This calculation is handled automatically and internally. This LP engine ensures mass balance, resolves zero time conflicts, and ensures that feedback loops are calculated accurately without propagation delays.

ExtendSim's discrete rate technology is fully integrated with discrete event simulation. Items can be converted into a rate (for example the arrival of a truckload of material) and back again.

### 4.3 Messaging

Messages are the fabric of an ExtendSim simulation. They are used to communicate information between the application and the blocks and from one block to another. Messages are sent to blocks at specific model events (such as model initialization), travel over connections, or are sent directly from one block to another. Messages ensure that each block has the most up-to-date information and that blocks update at appropriate times.

A message is interpreted by a block within the context of that block. For example, the clear statistics message causes a different set of actions in a queue than in an activity.

Without messaging, other simulation engines must rely on checking conditions at each step of the simulation, slowing performance.

### 4.4 Development Environment

ExtendSim uses a dedicated, compiled programming language for block development called ModL. There are numerous advantages to an embedded development environment that are specific to simulation modeling:

- **Interactivity.** Building an interactive model using a general purpose language requires that any interactive parameters be predefined and programmed into the model. In ModL, a change to a dialog parameter automatically becomes a change to a variable in the underlying ModL code. In addition, the dialogs that contain the result values are updated as the simulation runs.
- **Performance.** The additional overhead required to call an outside program such as a routine written in Visual Basic or C++ can have a significant impact on performance. In addition, in other compiled simulation programs there is a compile phase that must be done every time the model is changed. In ExtendSim, the blocks are precompiled, and recompiling not necessary.
- **Additional simulation-based functionality.** Generic programming languages have no inherent simulation capabilities. Functions and features have been added to ModL that are specific to continuous, discrete event, and agent-based simulation.
- **More robust programming environment.** ModL performs additional error checking and includes built-in data structures such as queues and linked lists that are not present in general purpose languages.
- **Integrated debugging.** The ModL debugger is built into ExtendSim. ExtendSim's system variables and function calls are visible from within the debugging environment.

- **User interface development.** The ModL language is "live" from the time that ExtendSim is started to when the application is closed. Because of this, ModL can be used whether or not the simulation is running. This is invaluable for creating user interfaces that are used before and after the simulation run as well as during the simulation run.

When compared to a simulation program that generates code (either in a simulation specific or general purpose language), ExtendSim's technology eliminates the need for the time consuming linking and compiling steps. Also, by definition, code generators can not include the level of interactivity provided by ExtendSim.

Note that ExtendSim does have sophisticated methods for utilizing subroutines, methods, and programs developed in outside languages. This can be done through the COM or DLL interface. However, because of the power and flexibility of ModL, it is rarely necessary that the modeler would need to resort to an external language.

## 5 APPLICATIONS

The range of applications where Extend has been used is truly impressive. Extend has repeatedly proven itself capable of modeling large complex systems.

### 5.1 Standing Joint Force Headquarters (SJFHQ)

This model simulates the information flow for 377 processes and sub-processes performed by the 58 people supporting Regional Combatant Commanders and/or Joint Task Forces. From the text of the paper: "Responsibilities for the SJFHQ are to maintain and provide information databases, collaboration capabilities, and subject-matter experts (SMEs) to assist in conducting the Effects-Based Planning (EBP) processes. SJFHQ personnel provide a core of trained experts who are essential to conduct planning using the new EBP processes".

**Error! Reference source not found.** shows the top level of the model. Each circle labeled "P" is a multi-level hierarchical block that represents a person in the model. These hierarchical blocks include details for simulating the multi-tasking activities required by all personnel.

The Extend version 6 database was used to store the specifics of the information flow for each scenario modeled. This allowed the modelers to change inputs from one place, run the model and view the results in the same central location.

### 5.2 Modeling of CD4+ Memory T Cell Generation

To those who are primarily simulation modelers with traditional industrial engineering or operations research backgrounds, this model is certainly something outside the normal realm of experience. However, it is an excellent

example of how Extend has been used for a new application of discrete event simulation.

The interaction of  $10^2$ – $10^8$  simulation items (lymphocytes) and their actions of dividing, resting, changing, and apoptosis are simulated in this model (Zand, 2004). These simulations closely mimic the biology of lymphocyte activation and expansion, in which mitosis vs. apoptosis decisions and division times are stochastically determined by individual cells. With this simulation model, large scale statistical results were obtained that could not be measured in a study with human volunteers. One advantage of discrete event simulations is that estimates of the absolute number of apoptotic cells arising from the CD4+ T cell response could be calculated. These values are not available from studies with human subjects. The simulation model also allowed more latitude for sensitivity analysis.

### 5.3 Biological Drug Production

Developing pharmaceutical and biotech products requires a significant capital expenditure even for large companies. Profitability in manufacturing these products is paramount in recovering development costs.

The biological production model in Figure 9 was developed for a pharmaceutical company by OpStat, Inc. It models the production of a biologically based product. This simulation captures the material flow, product changeovers, and staffing skill levels and availability. Results indicated a number of facility characteristics that would not have been uncovered without simulation modeling. These included a requirement for 24/7 staffing, material handling constraints, and the effect of variability in cleaning operations on throughput.

## 6 SUMMARY

ExtendSim 7 capitalizes on the success and experience of earlier versions. With a range of applications from T Cells to supply chains, ExtendSim has proven itself to be a robust and approachable simulation environment. With the largest professional customer base in the industry, Imagine That's "build a better mousetrap" business model has benefited simulation modelers worldwide.

## REFERENCES

- Angelides, Demos C. 1998. *Material Supply Simulation for Offshore Pipelaying Construction*. In Proceedings of the 2006 Winter Simulation Conference, ed. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, 1297-1304. IEEE, Piscataway, NJ.
- Hamber, Robert. 1999. *CloaDS & TloaDS*. 1999 Simulation Solutions Conference. Institute of Industrial Engineers, Norcross, GA
- Hutchins, Susan G., Gordon E. Schacher, James Dailey, John P. Looney, Steven E. Saylor, Jack J. Jensen, John S. Osmundson, and Shelley P. Gallup, 2005. *Modeling and Simulation Support for the Standing Joint Force Headquarters Concept*. Available via [http://www.dodccrp.org/events/10th\\_ICCRTS/CD/papers/366.pdf](http://www.dodccrp.org/events/10th_ICCRTS/CD/papers/366.pdf) [accessed June 29, 2007]
- Imagine That, Inc. 2001. *Extend User's Guide v5*. San Jose, CA.
- Krahl, Dave. 2001. *The Extend simulation environment*. In Proceedings of the 2001 Winter Simulation Conference, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 217-225. IEEE, Piscataway, NJ.
- Kress, R, Bills, K, J Dixon, and R Rinehart. 2006. *Database-Intensive Process Simulation at the Y-12 National Security Complex*. In Proceedings of the 2006 Winter Simulation Conference, ed. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1989-1996. IEEE, Piscataway, NJ.
- Pritsker, A. A. B., and C. R. Standridge. 1987. *TESS: The Extended Simulation Support System*. New York: Halsted Press.
- Zand M. S., Briggs BJ, Bose A, and Vo T. 2004. *Discrete event modeling of CD4+ memory T cell generation*. J Immunol 173(6):3763-3772

## AUTHOR BIOGRAPHY

**DAVID KRAHL** is Vice President of Technical Sales with Imagine That, Inc. He received a MS in Project and Systems Management in 1996 from Golden Gate University and a BS in Industrial Engineering from the Rochester Institute of Technology in 1986. Mr. Krahl has worked extensively with a range of simulation programs including Extend, SLAM II, TESS, Factor, AIM, GPSS, SIMAN, XCELL+ and MAP/1. A few of the companies that Mr. Krahl has worked with as a consultant and educator are Chrysler, Ford, Williams International, Tefen, Raytheon, and Boeing. He is actively involved in the simulation community. His email address is [davek@extendsim.com](mailto:davek@extendsim.com) and the Imagine That Inc. site is [www.extendsim.com](http://www.extendsim.com).

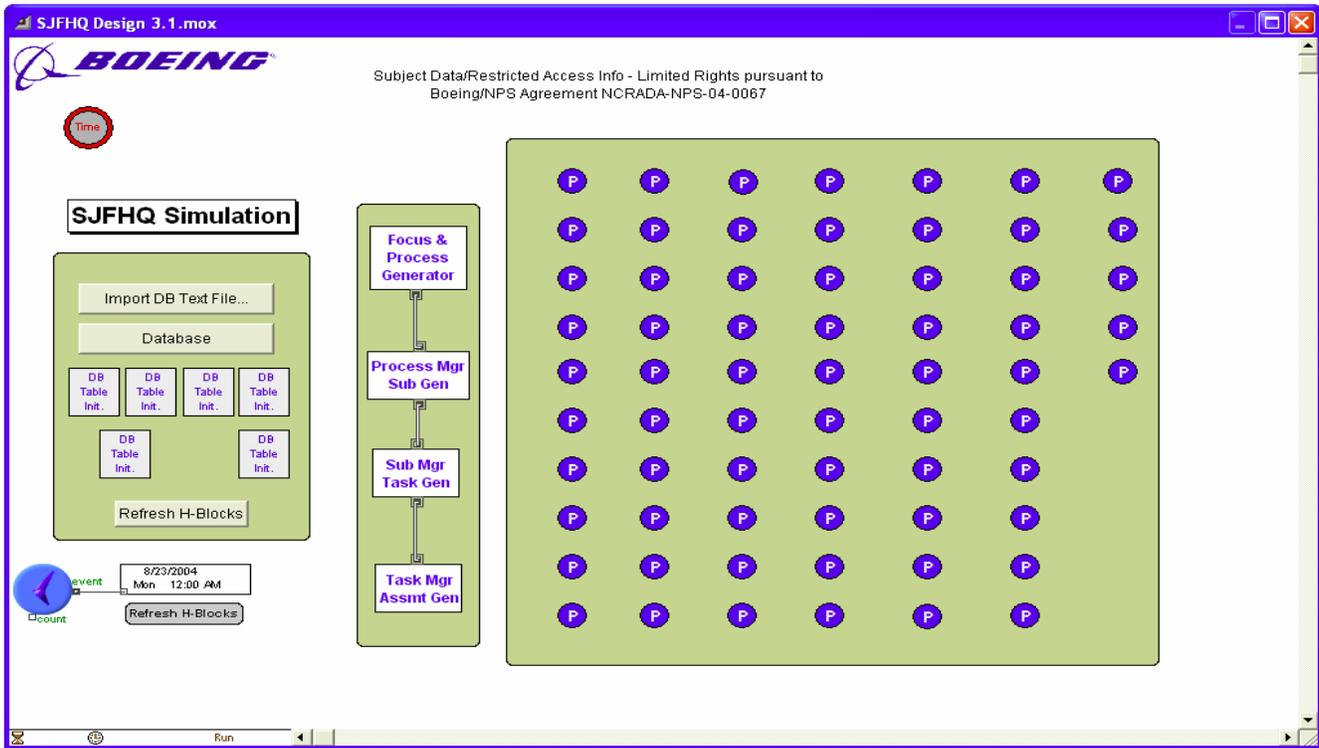


Figure 8: SJFHQ Model

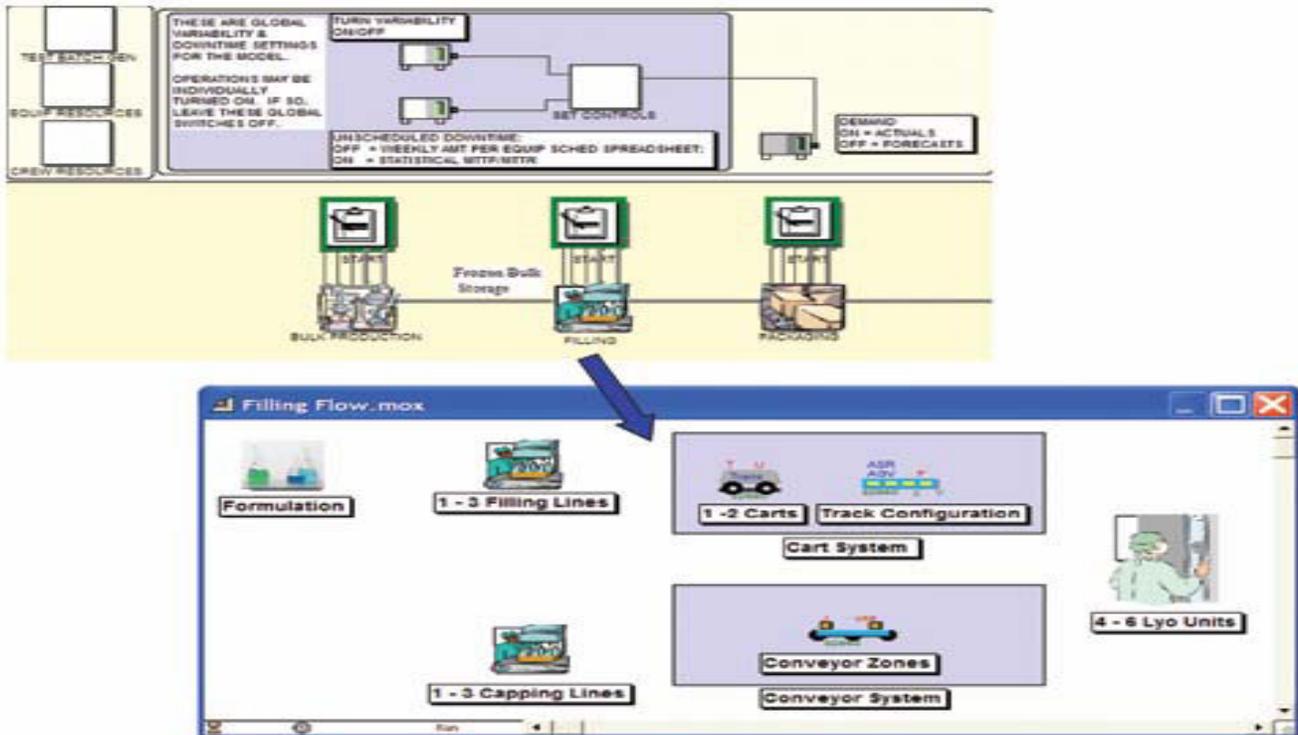


Figure 9: Portion of Biological Production Model