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

EXTENDSIM ADVANCED TECHOLOGY: INTEGRATED SIMULATION DATABASE

Bob Diamond
David Krahl
Anthony Nastasi
Peter Tag

Imagine That, Inc.
6830 Via Del Oro
Suite 230
San Jose, CA 95119, USA

ABSTRACT

ExtendSim is used to model continuous, discrete event, discrete rate, and agent-based systems. This paper will focus on ExtendSim's tightly integrated simulation database which provides features that facilitate database-centric modeling and improve and streamline the modeling process.

1 INTRODUCTION

A simulation model that is purposefully built around a central, integrated database requires less time to build, modify, maintain, validate, and verify. In addition it will tend to have fewer errors and be easier to incorporate model scale and data changes. Models built in this way begin with defining the structure of the required data. This has an additional benefit of moving the most common bottleneck of the simulation process, data collection, to the beginning of the project.

This modeling approach is facilitated by ExtendSim's Graphical Simulation Database (GSDB). This database system is accessible to every ExtendSim simulation component. It contains many features not found in ordinary databases. In addition to the typical database capabilities, ExtendSim's GSDB includes a number of features that uniquely support the requirements of a simulation model. These include data-linking, link-alerts, database aware modeling components, database address attributes, and database initialization options.

Even if the modeler does not explicitly use the database, ExtendSim's modeling components record information and model state in database tables. ExtendSim's Resource Manager, for example, records the properties, state, current allocation status, groups, and pools in database tables. These are manipulated during the modeling process and the simulation run to manage all aspects of the advanced resource feature.

2 EXTENDSIM STRUCTURE

An ExtendSim model is created by adding blocks to a model worksheet, connecting them together, and entering the simulation data. Each type of block has its own functionality, dialog, help, icon, and connections. Each instance of a block has its own data. Blocks perform a number of functions in a simulation model including:

- Simulating the steps in a process (Queue, Activity)

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

- Performing a calculation (Math, Random Number)
- Interfacing with other applications or data storage (Read, Write)
- Providing a model utility (Find and Replace, Count Blocks)
- Plot model results (Plotter, Histogram)
- Tools for interface creation (Popup, Buttons)

The logical entity that moves through the system is referred to as an item. Items carry properties or attributes with them as they progress from one block to the next. Items are represented using lightweight data structures allowing virtually unlimited numbers to exist simultaneously within a model.

The source code for all of these blocks is available and can be viewed or modified by the end user. Blocks can be created from existing blocks or created from scratch.

Hierarchical blocks contain other blocks (either programmed or hierarchical). This helps to organize the model. Hierarchical blocks can be stored in a library and reused in the same or different models. ExtendSim has tools for creating an interface within the hierarchical block, making it easy for the modeler to expose important parameters and results.

3 DATABASE AND SIMULATION HISTORY

Simulation and databases are not new. Early attempts at combining the two technologies were made by Pritsker and Associates in the 1980's (Standridge and Pritsker 1987). However, the technologies and platforms of the of this time period limited the usefulness of these products. Few models were built that took advantage of the database features provided with Pritsker's products. The first ExtendSim (then known as Extend) database was developed by Simulation Dynamics as an add-in (Phelps, Parsons, and Siprelle 2000). This proved to be vital in learning how databases can and should be used in simulation models. Because this database was external to the ExtendSim program, there were limitations on what features could be implemented and the overall level of integration. With version 7 of ExtendSim, Imagine That Inc. introduced a fully integrated, graphical database. This database is part of the ExtendSim program, not an external component or add-in.

4 MODELING OBJECTIVES AND DATABASES

Improvements to simulation technologies are driven by the needs and objectives of the users. These users fall into two general types: 1) the decision-makers who use the results from simulation models and 2) the modelers that build simulation models. Databases are a technological improvement that helps satisfy many of the objectives of both types of users. Decision-makers need results turned around more quickly and presented in convenient and meaningful formats. They also want accurate models that can be applied to a broader range of problems. In order to satisfy the objectives of decision-makers, modelers focus on finding ways to streamline and improve the modeling process. To do this, they need simulation technologies that reduce the time required to:

- Translate a problem into a simulation model
- Collect data
- Translate collected data into a form required by a model
- Verify and validate models
- Convert model results into a meaningful format for decision makers
- Maintain models

Modelers also require simulation technologies that allow them to design more flexible models that can be easily reconfigured to accommodate different types of problems.

To see how a database can help achieve these objectives, it is helpful to break the modeling process into a set of discrete stages as shown in figure 1.

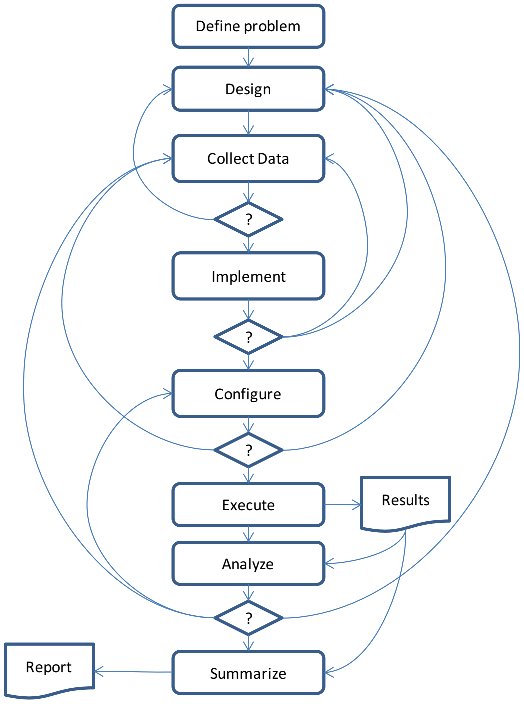

Figure 1: Modeling process stages

Each stage provides a different set of opportunities for streamlining and improving the modeling process. Having an integrated database allows modelers to design a simulation model with a data-centric point of view. This perspective is valuable because it provides opportunities to improve the simulation modeling process at multiple stages. After the problem has been defined, the next step in building a model with this approach is to design the database. This is done by identifying the important system entities, their properties and relations and translating these into database structures.

When the specifications for the simulation model are represented in database tables, it provides an opportunity to directly engage non-simulationists in the design stage of the modeling process. This can increase the applicability and acceptance level of the model. By structuring the database early in the design stage, it is also easier to determine what data is needed and communicate the requirements and format to the data providers. This reduces the time spent during the data collection stage which is often a bottleneck in the modeling process. Data defined and developed in this way will tend to be more complete and appropriate for the simulation model, reducing time spent iterating between the implementation, data collection and design stages. Having a well defined data design at the outset of the project will also reveal opportunities for creating smaller, more elegant and more scalable simulation models. This will increase model reusability and reduce the time required to maintain and reconfigure models.

At first blush, the only database features that are necessary to support simulation are tables, relationships and methods for getting and setting the data in the tables. These features improve and reduce the time spent in many of the modeling process stages. However, in order to fully harness the power of a database, the database components need to be tightly integrated with the other components in the simulation environment. The tighter the integration, the more streamlined the modeling process becomes. When the database becomes part-and-parcel of the simulation model, data redundancy is reduced, verification and validation time is shortened, the capability of the simulation environment is expanded, and models are more accessible to non-simulationists.

5 IMPLEMENTATION

When Imagine That Inc. set out to implement a next generation database in ExtendSim, the objective was to realize the full promise of database-centric modeling. In order to achieve this objective, they identified several advanced features that supported this vision:

- Rapid data access
- Data linking
- Database aware modeling components
- Embedded distributions
- Link alerts
- Database address attributes
- Database initialization

Since ExtendSim modeling is done with a graphical user interface, the user experience demands that databases be designed and accessed with that same ease of use, along with a suite of API (application programming interface) functions to allow blocks to create and access databases on the fly.

With the above features in mind, add-in and component-based databases were evaluated for ExtendSim's GSDB during the development processes. However, it was not possible to modify these off-the-shelf development tools to effectively implement the above features in a truly integrated simulation database. The GSDB is written in C++ and is part of the ExtendSim application.

5.1 Basic Database Functionality

As a baseline, any database used for simulation should have the following basic features. These are fundamental requirements for storing, accessing, and organizing simulation data and do not require additional descriptions or justifications.

- A variety of field types including string and numeric fields
- The ability to establish relationships
- Infinite "users" or access points to the database
- An interface for accessing the data from the model

5.2 Rapid Data Access

The GSDB has been optimized for rapid data access. Approximately 2 million reads and writes can be performed in a single second (Intel® Core™ 2 Duo T8400). The ExtendSim GSDB resides in memory to maintain high levels of performance. This kind of speed is essential for simulation models where the database is accessed many millions of times during a single simulation run.

The low computational overhead that occurs with database accesses in ExtendSim encourages modelers to represent as much of the system as possible in the database. This promotes better model design, increases model organization and streamlines the modeling process.

5.3 Data Linking

Data linking is the bidirectional sharing of values between the database and block parameters or tables. Whenever a linked value changes, all of the components linked to that value are modified as well. This occurs whether the model is running or not. Typically a model parameter is linked with the point-and-click interface, but it can also be done programmatically through the database API. Databases must support an infinite number of read/write sources with no-lockout to data access. Since there can be many read/write links to the same data, a block can access and even change the data at any time, even during changes initiated from another source. Linking eliminates data redundancy, reduces data errors, adds to model transparency, and assists in model organization.

5.4 Database Aware Modeling Components

There are a variety of methods for accessing the database beyond linking. Imagine That Inc. has created blocks that allow the modeler to get and set database values with ExtendSim's point-and-click interface. In addition many of the blocks have specific functions that automatically utilize database structures. For example, the Scenario Manager automatically creates database tables with the scenario factors and responses and the Resource Manager uses the GSDB to store all of the information relevant to the Advanced Resources. These blocks, by leveraging the ExtendSim GSDB, provide a superior functionality and user interface experience making it easier to access and work with the data for the model.

5.5 Embedded Distributions

Database cells can be specified as random distributions. Each time the cell is accessed, a value is sampled from the distribution. Named distributions can be created and used in multiple database cells. When one cell can provide a complete random distribution, it allows places in the model that require randomness to quickly access a statistical distribution from a single location in the database. This allows the modeler to maintain the simulation input data as a distribution rather than using separate database fields to represent the distribution type and arguments.

5.6 Link-Alerts

Perhaps the most important feature of ExtendSim's GSDB is its embedded messaging capability. Termed "link-alerts" these messages can be used for any type of state-change notification. When a value changes in the database, blocks that are dependent on that value can receive a message or notification that the value has changed. The ExtendSim GSDB supports a lightweight reentrant notification to all dependant model components.

Link alerts provide real-time notification of system state changes, eliminating the need for polling and redundant calculations. Link-alerts eliminate problems resulting from data latency.

5.7 Database Address Attributes

In addition to number and string attributes, ExtendSim's items can take on database address attributes. These point to a database location: database, table, field, or record (or combination of these). This makes it easy to set up shared or extended item properties. For example a database attribute could point to a routing table shared by all items with the same type or it can be used to attach a unique table that contains the item's historical information.

Database address attributes provide a more compact method for accessing and maintaining complex structures. This reduces model overhead and improves model scalability.

5.8 Database Initialization

The GSDB is used for a variety of purposes in the model. It can be used for simulation inputs, a working dataset, or to store model outputs. In each of these cases, there are specific database features that facilitate its use. For example each field can be optionally initialized at the start of every run. If the GSDB is used to store simulation results or working data, this saves the effort of manually initializing the field. Another example is the application level functions that can import an entire table from an external application, such as Excel, into a database table in one step. With this technology, large datasets can be imported in a fraction of a second.

Database initialization provides modelers with a number of shortcuts to perform commonly used simulation operations that don't exist in other database technologies. This reduces the time to build and configure models.

5.9 Sample Usage of GSDB Features

To provide a more concrete example of how these features are used in actual modeling, a simple, theoretical inventory model has been assembled. In this type of inventory model products are purchased from inventory. When the inventory falls below a reorder point, additional product is ordered to be produced.

The simple model shown in Figure 2 displays how ExtendSim's GSDB features impact the model. At the start of the simulation inventories are initialized with ExtendSim's database initialization features. When a purchase occurs, the new inventory level is written to the database. Because the database is optimized for performance, this requires little overhead for the simulation model calculation. When the change occurs, the linked inventory manager receives a message. If the conditions warrant, the inventory manager can react to the inventory change, placing an order to production. The order consists of an item with a database attribute pointing to a bill of materials (BOM) table. The production delay is calculated from a database table containing random distributions.

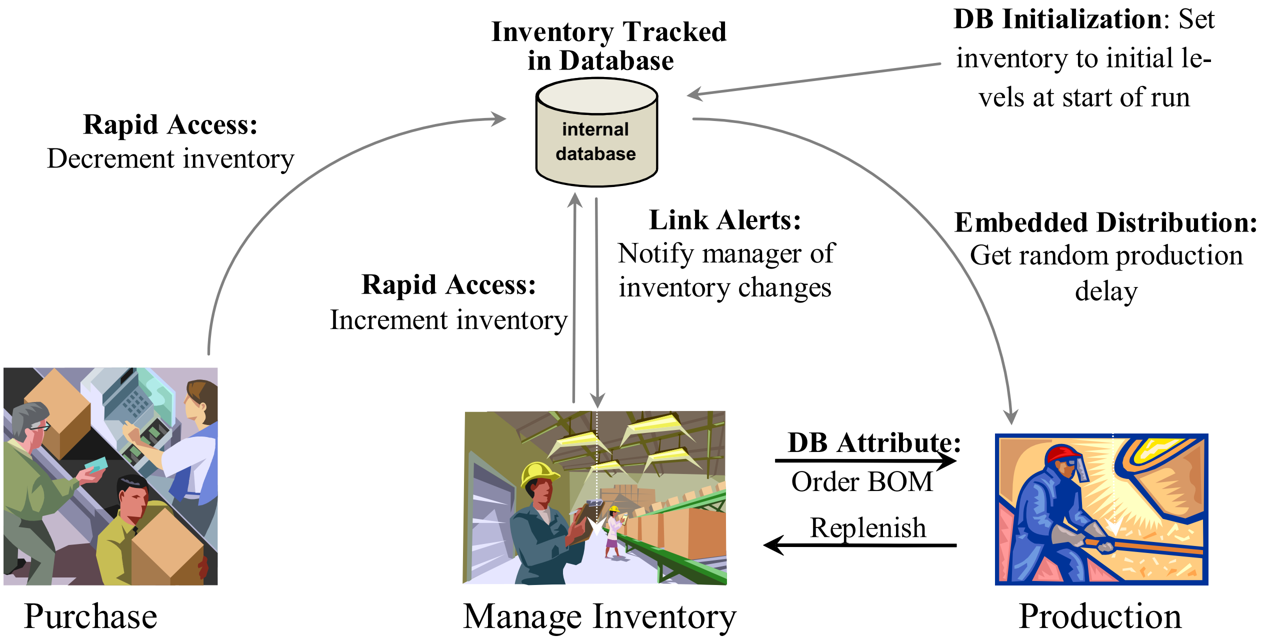
Inventory Tracked in Database
internal database
DB Initialization: Set inventory to initial levels at start of run
Rapid Access: Decrement inventory
Link Alerts: Notify manager of inventory changes
Embedded Distribution: Get random production delay
Rapid Access: Increment inventory
DB Attribute: Order BOM
Replenish
Purchase
Manage Inventory
Production

Figure 2: Sample inventory problem

6 SUCCESS STORIES

Extendsim's database is widely used by Imagine That's extensive customer base. The most sophisticated ExtendSim modelers use the database as the core of their simulation models. This has allowed them to be more productive, delivering simulation models of complex systems on time and within budget.

6.1 Healthcare

FDI Simulation (Faller, Flynn, and Ferrin 2009) was able to create a single model that simulated different prison healthcare systems. The GSDB was used to contain the data specific to each facility. Changing the model to a new prison only required importing a different database. They were able to efficiently simulate over 350 different scenarios eventually developing a new process which was superior to the current processes at all of the modeled prisons.

6.2 Agent Based Simulation

The Dynamic Security Airport Simulation (Weiss 2009) is an agent based model that simulates various path-weapon combinations and their interaction with airport security services. ExtendSim's database is used to centralize the user-defined parameters for the staffing, sensors, and threats. For example, tables of detection of each weapon by type were created. A user interface control panel was also created with frequently changed parameters.

6.3 Supply Chain Modeling

The Boeing BALANCE model (Saylor and Dailey 2010) illustrates enlightened use of ExtendSim's GSDB. The Boeing Advanced Logistics Analysis Capabilities Environment (BALANCE) is an accurate highly user configurable system for simulating complex multi-echelon, multipart indentured supply chain networks. This modeling system includes capabilities for simulating subassemblies and refurbishment. In the BALANCE environment, simulation data may be manually or automatically configured using an auto-build capability. This modeling system can be used across a wide range of supply chain and logistics domains. *"ExtendSim was selected as the basis for development due to its tightly integrated data-base architecture, underlying messaging architecture, and custom block development environment that easily facilitates development of customized data-base driven simulations."*

7 CONCLUSION

ExtendSim has a variety of technologies for simulation modeling. A more general overview can be found in the 2008 Winter Simulation Conference Proceedings (Krahl 2008), the GSDB is only one of them. It is, however unique in its implementation and capabilities. With effective use of ExtendSim's integrated database, simulation processes can be streamlined, producing more reliable, scalable, and flexible simulation models.

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.

Faller, S., C.T. Flynn, and D. Ferrin. 2009. Simulating healthcare in prison systems. In *Proceedings of the 2009 Winter Simulation Conference*, M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin and R.G. Ingalls, 2032-2041. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Phelps, R., D. Parsons, and A. Siprelle. 2000. The SDI industry product suite: simulation from the production line to the supply chain. In *Proceedings of the 2000 Winter Simulation Conference*, J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick, 208-214. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Saylor, S., and J. Dailey. 2010. Advanced Logistics Analysis Capabilities Environment. In *Proceedings of the 2010 Winter Simulation Conference*, ed. B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan. To appear. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Standridge, C.R. and A.A.B. Pritsker. 1987. *TESS: The Extended Simulation Support System*. New York: Halsted Press.

Weiss, W. 2008. Dynamic security: an agent-based model for airport defense. 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, 1320-1325. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

BOB DIAMOND is the creator of the ExtendSim simulation framework, database, and the founder of Imagine That Inc. Before ExtendSim, Bob created the DesignScope simulation application used in signal processing. While in school, Bob was recruited by NASA Apollo to create the Fuel-drop simulation of combustion within the Saturn booster engine. As Principal Scientist at Commodore Intl., he used simulation extensively to validate new technology during the birth of the personal computer. At Consolidated Video Systems, he designed broadcast digital video signal processing hardware using simulation and was the recipient of a Rockefeller/Ford Foundation grant for his work as technologist/artist-in-residence at WNET/NY. His multimedia collaborations with Fluxus artist Bob Watts and composer David Behrman have been shown at the Whitney in NYC and in other venues worldwide. His email address is <bobd@extendsim.com>.

DAVID KRAHL is Vice President of Technical Sales with Imagine That Inc. He received an 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 ExtendSim, 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> and the Imagine That Inc. site is <www.extendsim.com>.

ANTHONY NASTASI is a simulation software engineer at Imagine That Inc. He received a Masters of Science in Regulatory Economics from the University of Wyoming. Before joining Imagine That Inc. in 2000, Mr. Nastasi worked 4 years at Los Alamos National Laboratory modeling textile supply chains. He can be contacted by email at <anthonyn@extendsim.com>.

PETER TAG is a simulation developer with Imagine That Inc. He received an MS in Geophysics from Cornell University in 1979 and a BS in Geological Sciences from Lehigh University in 1977. Some of the simulation programs Mr. Tag has used include ExtendSim, iThink, Vensim, ProcessModel, AutoSched, AutoMod and G2. Mr. Tag has worked as a simulation developer and consultant for several different companies including Intel, Boeing, Procter & Gamble, CitiCorp, Monsanto, Johnson & Johnson, Raytheon, Los Alamos National Laboratory, Milliken & Company, and Pratt & Whitney. His email address is <petert@extendsim.com>.