Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

EXTENDSIM: A HISTORY OF INNOVATION

David Krahl

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

ABSTRACT

ExtendSim defined the modern simulation software environment. The structure and features that were put in place in the late 1980's are now commonplace throughout the simulation industry. The overall architecture of the first version of ExtendSim was so robust that it remains the foundation of the current generation of ExtendSim products. This feature set continues to lead the industry in scalability, ease of use, and interactivity.

1 INTRODUCTION

ExtendSim (originally named "Extend") was the first simulation software to incorporate the features found in modern simulation environments. Its user interface, architecture, and extensibility set a standard that has been followed by numerous other simulation programs. Before ExtendSim, building a simulation model involved programming in a simulation language - writing extensions in FORTRAN or C, changing parameters in a text file or the body of the simulation model, submitting the simulation run, and then waiting for the completion of the run before getting any idea of the performance of the simulation model. If it was a medium or large model, all of this had to be done on a minicomputer or mainframe. ExtendSim introduced interactive, drag-and-drop modeling, and a development environment for creating unique, reusable modeling components. All of this, including unlimited model size, could be done on a desktop computer.

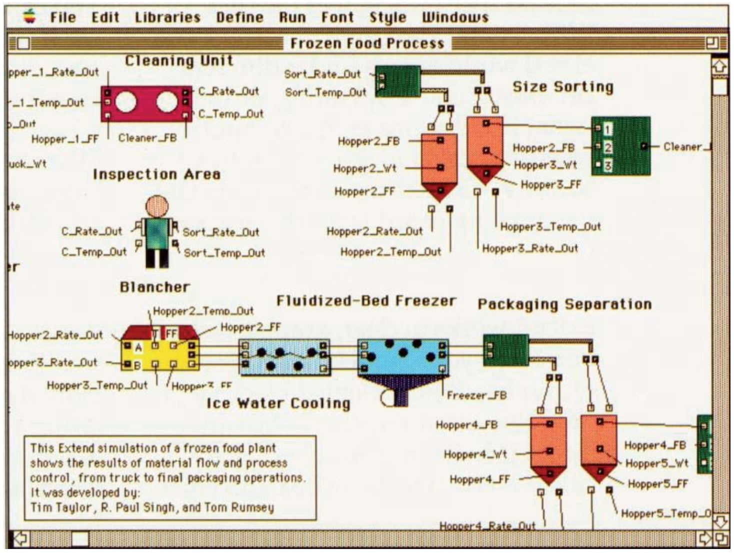
2 HISTORICAL PERSPECTIVE

In 1988, Windows 2.1 was the current graphical user interface from Microsoft (although it had yet to gain any significant traction) and the vast majority of PC applications continued to be limited by DOS's 640 kilobyte limit restricting simulation model size. SLAM II, Siman, GPSS, and Simscript were popular simulation programs and considered state-of-the-art. At that time companies typically had one or two simulation application experts who worked on minicomputers or mainframes (MacWeek 1989) and all modeling projects were funneled through them.

1988 marked the release of ExtendSim 1.0. For the first time, simulation modelers could enjoy building models with drag-and-drop ease, creating their own modeling components, and operating the simulation interactively. ExtendSim was also a paradigm shift in that the subject matter experts became the modelers and the simulation application became another desktop tool.

Other than the early Macintosh interface, Figure 1 (MacWeek 1989) illustrates how similar an early version of ExtendSim is to modern simulation software. Of course, over time, the interface has been fine-tuned, new features added and new operating systems supported, but the core design which caused such innovation in the simulation industry has essentially remained unchanged.

978-1-4673-4781-5/12/$31.00 ©2012 IEEE



This Extend simulation of a frozen food plant shows the results of material flow and process control, from truck to final packaging operations. It was developed by: Tim Taylor, R. Paul Singh, and Tom Rumsey

Figure 1: ExtendSim model from 1989

Compare the ExtendSim model with simulation software from the same time period (Pegden, Shannon, and Sadowski 1990) as seen in Figure 2. It’s easy to see why the simulation industry emulated ExtendSim.

```
BEGIN;
         CREATE: EXPONENTIAL(9):                 !create jobs
                 MARK(TimeIn);                   set entry time
         ASSIGN: JobType=DISCRETE(.3,1,1.,2):    !set job type attribute
                 Status=1:                       !set status attribute
                 Priority=JobType;               !set priority attribute
Merge    QUEUE, MachineQ;                        wait for the machine
         SEIZE: Machine;                         seize the machine
         DELAY: NORMAL(Mean(JobType,Status),     !delay by processing
         Std(JobType,Status));                   time
         RELEASE: Machine;                       release the machine
         QUEUE, InspectQ;                        wait for the inspector
         SEIZE: Inspector;                       seize the inspector
         DELAY: TRIA(5,8,10),JobType;            delay by the inspection
         RELEASE: Inspector;                     release the inspector
         BRANCH, 1:
         WITH,.8,Good:                           !good parts
         WITH,.1,Reject:                         !rejected parts
         WITH,.1,Bad;                            rework parts
Good     TALLY: JobType,INT(TimeIn):DISPOSE;     tally time in system
Reject   COUNT: Rejects:DISPOSE;                 count the rejects
Bad      ASSIGN: Status=2:                       !reset status attribute
         Priority=3:NEXT(Merge);                 reset priority attribute
END;
```

Figure 2: Typical simulation software model from 1990

3 THE INVENTOR

Bob Diamond's varied background and experiences came together in the creation of ExtendSim. He began programming simulation models in 1965 for NASA as an undergraduate student. His simulation of the path of a fuel droplet within the Saturn V rocket engine was critical in discovering that the nature of combustion is a chaotic process. While simulation and the programming it required continued to be his main interest, Bob's career also included stints as a video and multi-media artist (one of his works, Cloud Music, is in the process of being acquired by the Smithsonian), hardware designer for the Commodore PET computer, and designer of the first broadcast digital video equipment at Consolidated Video Systems (CVS) .

Bob's vision, coupled with his entrepreneurial attitude and his background in engineering, science, the arts, and simulation made him uniquely qualified to create a new generation of simulation software. Inspired by the user interface of the then new Macintosh computer, Bob realized that simulation could be more accessible if it were graphical. He developed the DesignScope simulation program. While successful, DesignScope lacked a critical feature - the ability for the modeler to extend the simulation program. Bob’s second simulation application, ExtendSim, included a development environment so that modelers in any field could create their own components. The ExtendSim compiler was another major innovation. At that time, no compiler existed that could be seamlessly integrated into a graphical simulation application. So in the typical Silicon Valley style of the time, Bob created one from scratch. The ModL language, based on C, and the ModL compiler have been enhanced over the years and remains one of the hallmark features of ExtendSim today. ExtendSim even earned Bob a “You grok” from Steve Jobs.

4 DISCRETE EVENT SIMULATION

To be fair, ExtendSim 1.0 did not include libraries of blocks for modeling discrete event systems. That did not occur until version 1.1 (Imagine That Inc 1989). A testament to the soundness of the original ExtendSim design, its discrete event capabilities were created within ExtendSim’s internal development environment. Attributes, event scheduling, and the accompanying data structures were all created using ExtendSim blocks and the ModL programming language.

The technical challenges of incorporating discrete event simulation into a flexible graphical environment led to another ExtendSim innovation: its message-based discrete event architecture (Imagine That Inc 1992, Krahl and Lamperti 1997). Because ExtendSim’s blocks were independent of each other and could be enhanced and new ones created by modelers, a robust system needed to be created for communication between the blocks. As items pass through them, the blocks use messages to communicate with other blocks, sending and requesting information as well as propelling the items through the model. This system, developed by Steve Lamperti, has provided the flexibility to model a wide range of systems by combining ExtendSim blocks together. This technology allowed modelers to assemble simulation models in a way that modeled their system without having to work within the rigid framework of a traditional simulation program. For example, item handling blocks that do not have a capacity can be placed between a queue and an activity. This was not possible in simulation programs of that time (and is still not possible in many simulation programs). Features for path prediction and item gating were included for superior flow control of the items moving through the model.

5 THE ECONOMICS OF EXTENDSIM

Another innovation of ExtendSim is a direct result of Imagine That’s passion for simulation. Imagine That believes that simulation can be used by a wide range of people who would be discouraged by the typical high cost of simulation software. So ExtendSim was made affordable. ExtendSim 1.0 was only $395. In 1992, including the discrete event and manufacturing libraries, the price was still only $990. This was less than 10% of the cost of less sophisticated simulation software. Business success has proven out the strategy. Imagine That has always been and remains profitable with no debt since its inception and no outside capital.

The combination of an inexpensive, easy-to-use, and powerful simulation program has led to the adoption of ExtendSim in a wide variety of industries. ExtendSim has been used for applications and by people who would not otherwise have invested in simulation technology. And it’s not uncommon for someone to investigate ExtendSim because of the low price and later realize that it is an elegant, powerful simulation program. In addition to the traditional modelers such as industrial engineers and operations research analysts, our customers include anesthesiologists, military personnel, medical researchers, environmental scientists, MBA's, surgeons, IT analysts, architects, a variety of engineering disciplines, and computer scientists. And the uses of ExtendSim are just as varied: from medical research to satellite data processing to foreign policy. The Solutions Showcase (http://www.extendsim.com/showcase.html) and Case Studies & Papers (http://www.extendsim.com/papers.html) web pages contain a partial list of applications of the software.

6 SIMULATION THAT IS FUN AND ENGAGING

When ExtendSim was originally developed, simulation modeling was largely a text-based operation frequently requiring the use of FORTRAN subroutines to provide the necessary model logic. By creating an interactive drag-and-drop simulation environment, ExtendSim, for the first time, made simulation modeling fun. Modelers would be engaged by their model, encouraged to try new ideas because the process of building and operating a model was actually enjoyable.

6.1 Interactive modeling

ExtendSim has always been interactive. Parameters can be changed during the model run and the effects can be viewed immediately. No special effort is required for this; it's built into the architecture. Results can be viewed while the model runs. Each block reports its own statistics and these are updated as they are calculated. The modeler does not have to wait for the end of the simulation run to see the direction that the model results are taking. In many cases it is obvious shortly into the run that key results indicate poor system performance and the model needs to be adjusted. Even today, this kind of interactivity is the exception rather than the rule. While it is usually possible in other simulation programs to turn animation on and off during the run, simulation results are not typically available before the completion of the model run.

6.2 Fast turn-around

In ExtendSim, even though the individual blocks are compiled, no compiling step is necessary before a model run. Unlike the code generators still found in some simulation programs, compiling is done only for a particular block when its source code or structure has changed. And, unlike interpreted or managed code, ExtendSim's blocks are truly compiled for better performance. The modeler can go from running the model to making changes and back to running as fast as clicking the mouse.

6.3 Visual feedback

ExtendSim's animation is automatic and informative. The modeler does not have to reserve a separate area of the model for animation and add animation-specific objects to see the operation of the model. A single button click turns the default animation on and off. Of course, a number of advanced features are available to customize ExtendSim for a particular animation.

6.4 Live interface

Introduced in 1992, the cloning feature is another unique innovation that allows modelers to create a custom interface by dragging and dropping model parameters and results to the model notebook (Imagine That Inc 1992). This becomes a live interface, responding to input changes and updating results as the simulation runs. This gives the modeler easy access to important model parameters and metrics before, during, and after the simulation model run.

6.5 Ad-hoc experimentation

The above features mean that the modeler can change something, run the simulation, try something else, run the simulation, and so on in a minimum of time. The ease and speed at which a modeler can try out different operations and see the effects of those changes makes the simulation experience more engaging and even fun. Engineers, analysts, and systems-oriented people are natural brainstormers wanting to try out new ideas and see their effects. ExtendSim's support for this thought process is unparalleled.

7 INTELLIGENT BLOCKS

At the time of ExtendSim's introduction there were no simulation tools that allowed the modeler to create their own modeling components. They were either a defined, compiled simulation language or the modeling components consisted entirely of calls to subroutines within the simulation application. This limited the flexibility of the simulation environment as it was impossible to create user-defined modeling components. ExtendSim introduced a unique environment where new compiled blocks can be created and seamlessly integrated into a model with existing blocks. Even now, in other simulation programs that allow the integration of user-defined modeling components, interpreted or managed code is the most common. While easier for software developers to create, this type of technology has inherently slower performance than compiled modeling components.

ExtendSim is based around the concept of blocks - intelligent, pre-compiled modeling objects that communicate with each other through a defined interface. Blocks perform all of the actions in a simulation including: event-list management, item processing, data import and export, results presentation, debugging, and animation. Each block contains the ModL code, icon, help, and dialog. These components reside in a library. Libraries are collections of blocks within a category. Modifications to any of components of the block in the library are automatically changed in all blocks of that type. Once a block is brought into a model worksheet from a library, the block contains data that is unique to the instantiation of that block. The data is independent of the block's structure and is maintained in the model.

Over the years, ExtendSim's blocks have taken on a life of their own. In some cases literally. Agent based simulations sometimes move the blocks around the model. This can simulate autonomous robots detecting other objects and nearby robots as they move, or elevators carrying people to the appropriate floor in an office building. Blocks have also been created that added blocks, removed blocks and set the parameters of other blocks in the model. This allows the modeler to dynamically create the simulation model based on the inputs of the model and the properties of the data.

8 TRANSPARENCY

A transparent simulation program is one where the model appearance reveals the model behavior. ExtendSim's blocks are designed to be discerned by modelers and non-modelers alike. The iconic representation uses a combination of flow-chart and intuitive symbols. Animation on the block's icon relays information about the status of the block. A modeler or subject matter expert can understand the basic logic by viewing the model layout.

Not only are the paths that the items take visible on the model worksheet, ExtendSim's value connectors show the logical relationships. The modeler can decide how best to represent the mathematical calculations within the model. Figure 3 illustrates two different ways to calculate the time required to transmit a message based on the message size and transmission rate. These are equivalent methods for the same operation. In case A, it can easily be seen that there is a lookup operation in the Read block. The result of this is divided by the data rate to get the delay. In case B, the modeler chose to do the entire calculation in an equation. The advantage of case A is that the user can very clearly see the relationship between the delay in the Activity and the size of the message. The advantage of case B is that the model contains fewer blocks and will be smaller and more efficient. The advantage of ExtendSim is that modelers have the choice to choose the best method based on their preference and the intended audience for the model. In either case, the set of blocks can be encapsulated in hierarchy (see section 9.1) so that the top level of the model does not become cluttered.

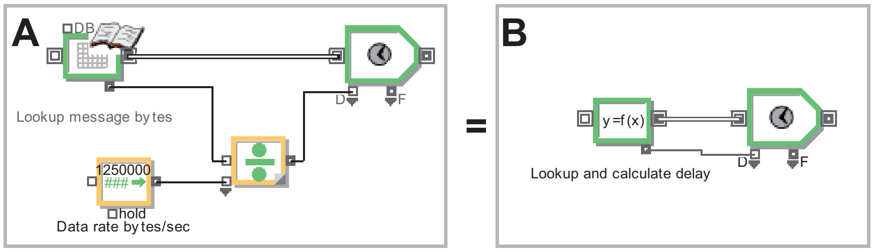


Figure 3: Delay is message size divided by bytes/second

Even today, to model a calculation such as a lookup requires programming in many other simulation programs. Ultimately the relationship between the inputs and outputs is hidden inside the structure of the modeling components. This makes it more difficult to communicate the operation of and to maintain or enhance a simulation model.

9 LARGE MODELS

A “large” model can be defined as one with many steps such as a chemical plant, sizeable data sets such as a supply chain, or complex logic such as models that simulate human decision making. Models created for real-life situations can contain many thousands of blocks, hundreds of thousands of records in data tables, as well as complex logical conditions. ExtendSim contains a number of tools for managing these large-scale models. The choice of the Macintosh operating system in 1988 not only provided a superior user interface experience, but Macintosh did not suffer from the same memory limitations as DOS based computers. In fact, the Macintosh processor (the Motorola 68000) was also found in minicomputers of that era. Eventually Microsoft Windows would mature to support 32-bit applications. In 1995 ExtendSim was made available for Windows-based computers (Imagine That Inc 1995).

9.1 Hierarchy

Organization of a graphically-based model can be difficult. Models can easily span multiple high-resolution computer displays. This makes the model intimidating to others and hard to manage on the part of the modeler. Introduced in 1992, well before any similar features were available elsewhere, ExtendSim’s hierarchy allows the modeler to divide the model into hierarchical sections. Hierarchical blocks are collections of existing blocks and include their own icon, help, set of member blocks, animation, and connections. They can be stored in a library and duplicated any number of times in a given model. Through the use of appropriate blocks, unique, permanent identities can be assigned to individual hierarchical blocks. This identifier can be used to differentiate the data in one hierarchical block from another.

9.2 Database

Simulation models are largely dependent on data. The management, access, and transfer of information to, from, and within a simulation model is a significant part of the model construction process. Certain classes of models such as supply chain and communication systems require large and potentially complex data sets to support the model. Early simulation databases in the late 1980’s were limited by the capacity and speed of computers at the time (Grant and Starks, 1988). As models have grown over time, so have their need for information.

ExtendSim first released a simulation-based database in 2001 (Imagine That Inc 2001). The database streamlines the management of model data. There are a number of differences between ExtendSim’s database and data management capabilities now found in other simulation programs:

- Integration – The database can be connected to the model in a variety of ways. There are specific modeling components for accessing data, individual dialog parameters and tables can be linked to a database table, equations have database-related input and output variables, and there is a set of database related functions. No other simulation program has an equivalent level of data management integration features.

- Speed – The ExtendSim database did not begin with an add-on or off-the-shelf database plug-in developed for general purpose use. Instead it was written specifically for simulation applications. ExtendSim's database is engineered for memory and performance efficiency within the context of a simulation model.
- Link-alerts – When a value changes in the database, blocks that are concerned with that specific piece of information are immediately notified. No end-of-event polling is required. This is handled internally and seamlessly by the application.
- Random numbers – An ExtendSim database cell can be specified as any one of a number of random distributions (including empirical). Whenever the cell is accessed, a sample from that distribution is used.

9.3 Equations

ExtendSim has a variety of equation-based blocks that allow the modeler to drop in logic at specific points in the model. Equations written in ModL are compiled when initially created or modified and do not re-compile at the start of the simulation run. This provides the efficiency of a compiler with the immediacy of an interpreter, the best of both worlds. Blocks that utilize equations can create model interfaces, run simulation models, execute database queries, perform mathematical calculations, animate models, and more. An equation debugger allows the modeler to step through the equation as it executes. A test button allows the modeler to evaluate the equation even when the simulation is not running. These features make debugging an ExtendSim equation a snap.

Elsewhere, simulation modelers continue to contend with writing simulation logic as external code in a programming language such as C++ or Java. While these languages are useful for software developers (ExtendSim is written in C++), they are burdensome for simulation modeling requiring separate compiling and code generating or linking steps.

9.4 Development environment

Combined with its open-source architecture, ExtendSim's development environment allows modelers unprecedented control and access to the internals of the simulation model. Everything from event scheduling to resource allocation is available for the modeler to inspect, verify, and even modify. Of course, this is not why simulation software is generally purchased, but it's reassuring to know that ExtendSim allows this kind of access to the simulation engine if necessary. To stay current, the development environment continues to evolve incorporating new features and technologies.

10 NEW TECHNOLOGY: DISCRETE RATE SIMULATION

In 2007, Imagine That pioneered a new simulation technology, discrete rate (Imagine That Inc 2007). Cleverly combining linear programming and discrete event simulation, this mixed analytical technology allows modelers to create accurate, efficient, and elegant models of rate-based systems. No longer does a model builder have to deal with the calculation errors and inefficiencies of a fixed step size or chunking the continuous flow into discrete items. The discrete rate capabilities of ExtendSim simulate rate-based and combined discrete event rate-based systems with unprecedented accuracy, ease, and performance. This technology remains unique to ExtendSim.

11 WHY IS THIS HISTORY IMPORTANT?

Truly game changing advances in technology are rare. Simply adding evolutionary features based on existing technology enhances the current state of the art, but it does not change the way that people think about or use a given technology. A real game changing product creates a new technology, changes the audience, or changes how a technology is used. ExtendSim opened up simulation to an entirely new class of modelers. By creating software that was uniquely powerful, accessible, and extensible ExtendSim found its way onto thousands of desktops. At a time when a large company might have a few copies of traditional simulation software, they were purchasing hundreds of copies of ExtendSim. Some companies provide each engineer with an analytical package that includes Microsoft Office, statistical software, and ExtendSim so that these tools would be available to them. Dramatically increasing the access to simulation modeling is the game changing effect of ExtendSim's technology. Imagine That puts this focus behind future versions of ExtendSim. We are proud of our heritage as an innovator and are committed to developing and improving our simulation software and supporting our customers in their modeling efforts.

12 CONCLUSIONS

ExtendSim changed the look, feel, and capabilities of simulation software. It introduced a new era of drag-and-drop modeling. It achieved this without sacrificing the capability to model large, complex systems. In fact, ExtendSim excels when modeling the most challenging problems. Perhaps ExtendSim's most important achievement is that its engaging interface made simulation fun, inspiring modelers who would not have otherwise used simulation to create models and enjoy the process.

REFERENCES

Cloud Music, http://www.vasulka.org/Kitchen/PDF_Eigenwelt/pdf/152-154.pdf, accessed July 11, 2012.

Damiron, C. and Nastasi, A. Discrete Rate Simulation Using Linear Programming. *In Proceedings of the 2008 Winter Simulation Conference*, Edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, 740-749. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Grant, M and Starks, D. A Tutorial on TESS: The Extended Simulation Support System. *In Proceedings of the 1988 Winter Simulation Conference*, Edited By M. Abrams, P. Haigh, and J Comfort. 137-140. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Imagine That Inc. 1989. Extend 1.1 Software Manual, San Jose, CA.

Imagine That Inc. 1992. Extend 2.0 Software Manual, San Jose, CA.

Imagine That Inc. 1995. User's Manual for Extend 3.x, San Jose, CA.

Imagine That Inc. 2001. Extend 5.0 User's Guide, San Jose, CA.

Imagine That Inc. 2007. ExtendSim 7 User Guide, San Jose, CA.

Macweek, March 21, 1989.

Pegden, C. D., R. E. Shannon, and R. P. Sadowski. 1990. *Introduction to Simulation Using SIMAN*, McGraw-Hill, Inc., New York, New York.

AUTHOR BIOGRAPHY

DAVID KRAHL is the Technology Evangelist 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, U.S. National Park Service, Idaho National Engineering Laboratory, United Technologies, 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>.