

# CHAPTER 10: THE GSS PARADIGM

The General Simulation System (GSS) represents the most significant advance in the development of modeling and simulation tools for many years. Featuring a Computer-Aided Design (CAD) environment taken from engineering, GSS eliminates the need to learn an esoteric programming language. Analysts and modelers can build models without the aid of programmers or programming experience. The easy-to-use CAD environment supports model development, scenario development, simulation, and analysis. Using windows, menus, and natural language, GSS users can quickly build models, run simulations and analyze results.

The underlying framework for GSS is based upon two totally new paradigms that depart from programming languages. These new paradigms allow one to build an architecture graphically with a one-to-one mapping from the symbols to the language elements. This provides visualization from the top of a system directly to a level of detail not achievable in software. This framework supports the development and evolution of very large models. Models can start small, grow, and be coupled into a single simulation or multiple simulations. Multiple simulations can run on one or more platforms simultaneously, synchronized to either the simulation clock or the real-time clock for live tests or demonstrations. According to PSI's international clientele, GSS houses the world's largest simulations.

GSS is known for its simplicity and ease of use. It does not take more than two to three days to be building models that represent precisely what the modeler desires. A discrete-event approach coupled with a distributed database architecture accommodates complex decision rules. This supports more accurate representations of nonlinear physical phenomena, eliminating the need for abstract nonlinear mathematical modeling. However, if one desires to represent systems using nonlinear mathematical models, GSS accommodates those requirements easily.

GSS has been used to model everything from complex coding and compression algorithms for communication protocols, to multiple networks of switches, gateways, routers and subscribers, to systems represented by differential equations, to models of human behavior, to biological models, to business markets, and to simple business decision systems. Large scale simulations run very fast allowing GSS optimization facilities to help modelers find the best model parameters to fit test data and accurately predict future outcomes. GSS provides for automatic multiple simulation reruns for parametric and sensitivity analysis.

Using GSS Run-Time Graphics (RTG), users can build dynamic graphical representations of simulation events while they unfold. ICONs can be made to move, take on different shapes and colors, and be interconnected by lines of different color, style and thickness. Various types of instruments can be inserted and meter readings updated dynamically. RTG provides for user interaction while the simulation is running. Users can change the simulation as it unfolds,

inventing trials and making decisions as to what causes various effects - interactively, and demonstrating the different results to other viewers.

## **FREQUENTLY ASKED QUESTIONS**

### ***What makes GSS an Open Environment?***

Models and simulations can be run on any of the popular UNIX platforms, e.g., SUN SparcStations, x86's or Pentiums running SOLARIS, SCO UNIX, or LINUX, the IBM RS-6000, and any SGI platform, e.g., the Indy, Indigo, Challenge, etc. And, it is now available.

Models written in FORTRAN, C or C++ can be called directly from GSS. GSS models can be called from other programs as well.

### ***What makes GSS a CAD environment?***

Computer-Aided Design environments used in engineering provide graphics and user friendly interfaces that eliminate the need for engineers and application experts to learn esoteric programming languages, e.g., C, and especially C++. These end users use the mouse in a point and click windows environment, graphically generating engineering drawings and natural language specifications that are structurally tailored to their environment.

GSS provides an architectural environment that uses engineering drawings to produce detailed designs of the models and their interconnections prior to use of any language. Hierarchically organized information structures are easy to describe and intuitively understandable by application area experts. Rules for changing the state of a system based upon system events are easily read and understood by anyone who understands the system. There is no need to learn a programming language. The environment is the same for any platform or operating system, eliminating the need to interface with UNIX or any other operating system. This totally eliminates the need for computer programming or system expertise.

The desire for CAD stems from significant improvements in both quality and economics. CAD eliminates the need to have a programming staff interpret what is wanted by the end user. This, in turn, eliminates misunderstandings in design requirements. It also eliminates dependency upon particular individuals who become irreplaceable because of their unique knowledge of the computer code they write. Both of these factors contribute significant time and cost savings.

### ***How do you support direct development of algorithms and software systems for production***

GSS provides for direct development, testing, and support of software algorithms and systems. Complete software modules and systems can be built using GSS, embedded in a GSS simulation, and tested using real-time scenarios. When these algorithms or systems are ready for transport to their production environment, they are automatically translated using the Virtual System Environment (VSE), that generates C-code for use on any platform having a C compiler. VSE and GSS are identical environments except for the discrete simulation scheduling facility used specifically for simulation. VSE modules and GSS models are directly interchangeable.

***What is the Visual Development Environment (VDE) option?***

GSS has two graphical support systems. The ***Run-Time Graphics (RTG)*** system is used to interact with the simulation - graphically - while it is running. The ***Visual Development Environment (VDE)*** is used to build model architectures graphically.

***What is the Run-Time Graphics (RTG) option?***

The Run-Time Graphics system (RTG) allows users to interact with the simulation graphically. While it is running, users can interact with a simulation and change models to the extent that they are constructed to support such interaction. For example, users can start with a bare map and deploy switches and radio units that can be turned on and start to operate. Radio units can communicate and form networks automatically, just like their live counterparts. Switches can be connected into networks and operate accordingly. Gateways between networks can be put down and communications routed across networks.

***How do you support multiple platform - Distributed Interactive Simulations (DIS)***

GSS has built-in facilities to support multiple platform simulations that allow users to interact with and affect results.