VisiSoft CASE STUDIES

This packet contains four case studies of prior projects where the use of VisiSoft provided a dramatic advantage.

VisiSoft CASE STUDY #1

PROBLEM REQUIREMENT

Develop a simulation to support the design, test, and evaluation of one of the world's (there are two) most complex mobile communications networks. In this network, all nodes are moving and connectivity between nodes is constantly changing. Most paths involve many hops, and alternate paths must be updated so that conversations can continue when a link goes down. Design of such networks cannot be done without huge simulations containing on the order of 500 nodes, each sending and receiving traffic to and from multiple sources simultaneously. The many complex software algorithms contained in this system must be represented in full detail.

PRIOR APPROACH

Three successive simulations were built by the prime contractor, with the first two ending in failure. The third time attempting to build it, 5 to 7 experienced engineers who had worked on the design of the communications system were put on the project. After about 4 to 5 years and $10M, it became almost impossible to work with the software and a 2 hour simulated scenario took 5 to 7 days. It was never fully completed. When the actual system was tested in the field, network build times were about 30 minutes versus the 3 minutes predicted by the simulation.

VISISOFT APPROACH

When given the task to rebuild it using VisiSoft, a team of mostly junior engineers completed it, going beyond the requirements, in about 18 months at a cost of about $750K. With calibrated models, subsequent predictions coincided with the field test curves. It was then modified to redesign the protocols to improve the network build and response times using a decentralized approach. When the requirements were still not met, fully distributed algorithms were built into the protocols with detailed dynamic graphics to see what was going on.

The resulting system was very large and complex, with nine layers of drawings that covered three walls. Design reviews were held at the prime contractors plant every 3 months at which we stood and reviewed the drawings. The prime contractor's engineers would ask questions pointing to the drawings, and we would answer them. No one looked at code. This took place between 1986 and 1990. This communication system would never have succeeded without the VisiSoft approach.

RESULTS

The resulting system, including facilities for building scenarios, two simulations, and data reduction and analysis facilities required about 1M lines of code when completed. It was considered the largest simulation of a communication system developed prior to the year 2000. The current largest (part of Case Study 2) is also built using VisiSoft.

VisiSoft CASE STUDY #2

PROBLEM REQUIREMENT

Network Planning Tool (NPT) - Develop a tool to support real-time planning and analysis for a communications network containing tens of thousands of nodes over national geographic areas. Provide a VisiSoft based planning tool to give the user a clean, concise and intuitive window into links and nodes type databases and their usage by their subscribers. The NPT must simulate a country's complete voice and data communication network and provide the ability to rapidly and accurately examine traffic flows. It must assess how a network supports its users, determine the criticality of a specific link or node, and analyze who talks to who and how.

PRIOR APPROACH

VisiSoft was chosen as the environment to develop the NPT over a prior system that was developed over several years, but was unsatisfactory in the following areas:

- Support for a limited number of nodes (hundreds vs. tens of thousands)
- Speed of execution (10 to 100 times slower than the requirement)
- Inability to visualize the network over its many layers of physical hierarchy
- Prohibitive Cost from annual license fees, excluding any product upgrades

VISISOFT APPROACH

The NPT was developed as an improved tool for visually depicting and analyzing large amounts of data. Network topology is drawn automatically on the geography of interest using links and nodes data. Interactive analysis of the ability for subscribers to communicate via a large network is supported. With this VisiSoft-based tool, an analyst can rapidly assess network vulnerability, and can selectively remove or degrade paths or links to determine the impact on subscribers. If degradations of links or nodes exist or are introduced, the simulation will reroute the connections using least cost paths until no connection can be found.

Analysts can zoom into an area of interest, and "pop" the cover on a facility to explore what's inside. VisiSoft provides the ability to push down to an arbitrary number of levels (no limit), and if needed, one can get down to the frame and cross-connect level. This means that analysis can be deep and exact, and allows vulnerabilities to be determined down to the individual wires. The Network Planning Tool is rich in its ability to deal both logically and graphically with complex networks and their databases. This client now owns a large collection of network models including satellites, ground stations, coaxial and optical cables, switches, routers, radios, distribution frames and junctions, all easily supported by the VisiSoft architectural facilities.

RESULTS

A very satisfied client who now has an extremely cost beneficial solution to his system requirement, and who has a modular framework for upgrades such as low-earth-orbit satellite communications, cell phones, packet switched networks, etc.

VisiSoft CASE STUDY #3

PROBLEM REQUIREMENT

In the 1980s, a Computer-Aided Design (CAD) system for developing discrete event simulations was becoming popular among clients. This CAD system made it much easier to build models of complex communications protocols and support the simulated testing needed to try new designs. It eliminated the massive software problem faced when evolving larger and more detailed simulations. However, behind the CAD system itself was a typical software nightmare that became almost impossible to enhance and debug.

PRIOR APPROACH

The CAD system, VSI's General Simulation System (GSS), provided users with a graphical architecture environment. This represented a paradox, since GSS itself was built using the latest OOP / C++ technologies. It typically took days to fix a single bug in the underlying GSS software - software that was hidden in a huge number of C-Code files and OS scripts for testing. People who best understood the requirements and underlying algorithms required to support discrete event simulation could not understand what was in the code. The top software person said it might take three months to find and fix a complex bug - with no guarantees. It was out of control!

VISISOFT APPROACH

VSI determined that a new software environment should be built based upon the principles that made GSS easy to use and understand. This was accomplished by designing new language translators using GSS, and slowly evolving the Visual Software Environment (VSE) from it. As VSE evolved, the architectures for GSS and VSE were totally redesigned. This created a much higher demand for VSI's own graphical CAD interface to the architecture; (prior to this point, third party CAD interfaces were used that were not well integrated).

As VSE evolved into a easy-to-use software environment, the goal of building a new CAD front end became realistic. In 2000, an initial CAD facility was created inside four months. This provided a significant leap in VSI's ability to build complex software that is easily supported and enhanced. The new CAD interface to both GSS and VSE makes software architecture visible, and thus a reality. New architectures are created rapidly and existing ones enhanced easily. This led to the VisiSoft product family that includes the Run-Time Graphics (RTG) system, with the Icon Library Manager (ILM), and Panel Library Manager (PLM).

RESULTS

The resulting VisiSoft system is built exclusively in VisiSoft. It is a huge complex commercial software system - with many user-friendly graphics and OS level functions - that spans millions of lines of code. Yet, it is easy to support and enhance.

VisiSoft CASE STUDY #4

PROBLEM REQUIREMENT

This military client wanted a tool to determine the best flight path for a specified set of platform parameters including on board sensors, self screening jammers, and electro-magnetic signals collection facilities. Also provided as inputs were red deployments and parameters for radars, SAM sites, target emitters, terrain, foliage and jammers. The basic optimization problem was to maximize a measure of prioritized collections while meeting constraints on probability of safe return of crew and platform, fuel, and other factors.

PRIOR APPROACH

The prior approach ran Monte Carlo simulations followed by single simulations with hand selected parameters. Each simulation took on the order of minutes to run, even though propagation models were neither optimized nor accurate. Setting up simulations took considerable time to enter way points with altitudes and speeds, and other deployment parameters.

VISISOFT APPROACH

A simulation was built that demonstrated the facilities contained in GSS and the ease with which large complex models could be revised and reused to solve this problem. Included were an accurate propagation model that is easily optimized for a given location; models of moving platforms with various equipment selection options, graphical inputs for flight path way points, altitudes and speeds; models of sensors, jammers, and SAM systems including communications between radars, C2 centers, and weapon systems; models of C2 decision processes, missile flyout, seek, lock-on, tracking, fuel capacity and consumption; and probability of hit.

Using the GSS optimization facilities, problem specification was satisfied with a complete and accurate set of models. Optimal flight paths were found using the automatic adaptive search algorithms built into the GSS optimization system.

RESULTS

Optimal flight paths were found that met constraints with typically less than a hundred simulations using the GSS adaptive search algorithms. These optimization runs typically took less than a minute for 100 simulations. Changing parameters, constraints, and optimization criteria was relatively simple and could be accomplished in minutes in an operational environment.