

# VisualSim Architect Aerospace

## System-level modeling and simulation software

### Key Features

- Validate and optimize the system design of electronics and timing-critical software
- Analyze across the three axes of function, timing and energy using dynamic models
- Construct large, complex systems using extensible library of modeling components
- Utilize the single environment for system design, functional validation and in-field debugging
- Export documentation from model content and simulate in a Web Browser

VisualSim Aerospace and Avionics solution encompasses a large library of parameterized modeling components, analysis tools and an extensive list of Application templates. Using these elements in the graphical VisualSim Architect modeling environment, engineers can assemble models of their radars, communications, navigation and avionics system, run Monte-Carlo simulation and generate reports that are used to optimize the product architecture.

Mirabilis Design is a Silicon Valley software company providing architecture simulation solutions for the exploration of large distributed electronic systems, semiconductor and timing-critical software. Mirabilis Design empowers design teams to create the right product using early architecture studies and analysis. Our product, VisualSim, is a modeling and simulation package that accelerates architecture exploration and enables the designer to create a product specification that meets the requirements.

The Next generation Bus specification team at NASA said, "Early results from in-house modeling activity of Serial Rapid IO using VisualSim indicate that the use of a switched, high-performance avionics network will provide a quantum leap in spacecraft onboard science and autonomy capability for science and exploration missions."

Using this modeling environment, users can assemble models of their proposed or existing systems. The designers can run simulations and conduct power, performance and functional analysis. Over 55 companies in Japan, US, China and India use VisualSim. Fully-trained technical experts are available in US, Korea, China and India to provide application and modeling support for our users.

VisualSim Architect is a modeling and simulation environment for system-level design of avionics sub-system, system interconnect, multi-board hardware and time-critical software. Using the pre-built parameterized library components, product development teams assemble sophisticated models of the proposed system or a concept, and conduct trade studies for a variety of requirements and budget constraints. The functional requirements are achieved with the constraints on timing, power, weight, and reliability.

VisualSim Architect has a well-proven methodology to facilitate design space validation. Based on the experience working with many large programs in the aerospace and avionics space, we have developed a step-by-step design process. Depending on your organization, location in the project schedule and type of design intent, the user may adopt one or more of the below methodology steps.

# VisualSim Architect

## Aerospace

1. **Blueprint:** A blueprint is created to identify the communications between all possible inputs and outputs, the sequence of actions in each flow, branch points and, anticipated delays for processing the task and retrieving data. Every system will have a series of periodic tasks such as a Sensor input, gyro or a software task, interspersed with aperiodic or non-predictable requests that are for data, events or other task completions.

The model will consist of traffic generators for inputs, sinks and plotters for the outputs and a set of expression, delays and schedulers for the system. This model is the foundation of the entire design process and can be used for multiple applications. The first is to identify the points of failure that can cause instant and delayed system malfunction.

The output provides the requirements for the development of redundancy systems and fault-tolerant handshaking mechanisms. Faults are injected into the system while it is in operation to measure the impact in each flow. Example of system faults include connector breaks, network node failure, incorrect data value, delayed data response, network congestion and timing jitters in task execution.

2. **Traffic and General Analysis:** This phase is used to optimize the system specification for lowest power consumption, detect errors caused by missing timing deadlines, impact of system faults, and create full system operation to debug problems in the field. Traffic analysis studies the impact of variable data patterns on a protocol or an implementation.

Synthetic traffic uses VisualSim libraries and real-life trace is from a prior or current system. The system will model the consequence of algorithm operation, delay for each algorithm task (high-level), and overhead activity, such as regular interrupts/RTOS querying/network requests. These traffic models can be used to define new or enhanced protocols and evaluated for different operation scenarios, traffic activity, and faulty operation.

3. **Architecture exploration** is the phase where the protocols are designed (MAC/Data/Transport layers, arbitration algorithms, flow management, etc.), study the data flow between sub-systems (check the sequence of operation, what happens when a sub-system fails, location of sensors, attenuators, mechanical subsystems, etc.), conduct hardware tradeoff (hardware component selection, etc.), software architecture selection (multi-threaded, distribution across multi cores and processors), and hardware-software partitioning (selecting the implementation mechanism based on target performance).
4. **Hardware and Software Modeling:** This is the architecture model that evaluates the detailed interaction between hardware and software. This model has three purposes-to size the hardware, distribute tasks across multiple processors/cores, and hardware-software partitioning. Here you can look at the detailed timing interaction- for example a multiple stage 1553B interaction with a Ethernet or the Timed of the Day clock schedule tasks against a random/alarm-driven execution. The tasks can be modeled at the data value level to evaluate the impact of late arrivals, incorrect values, reliability and fault-tolerance.

# VisualSim Architect Aerospace

5. Multi Domain Algorithm and Protocol Development: Every electro-mechanical system has control, analog, DSP and protocols. First these algorithms must be designed to meet the strict functional, timing and power constraints. There is a library of Regular Expressions, standard timing components, algorithm blocks, interfaces to network and multimedia devices, schedulers and queuing components that enable the assembly of these systems. Initially the detailed algorithms need not be defined. As more information becomes available, the details of the algorithm can be added to make the model have better fidelity and answer more implementation level details.
6. Software validation: Software validation is done after the code has been debugged. The system allows the user to validate the

safety, timing and the functional correctness of the software. Large systems have software on distributed terminals or nodes that interact across long, hierarchical and fast interfaces that make verification difficult without the assembled systems.

The architecture models are used to conduct functional, timing, safety and reliability analysis in the face of unpredictable errors and sensitivity to timing delays. This is critical when the result of one software task is needed by another periodic task, or when one task triggers another. You can study the reaction of the software to injected faults and determine system reliability. Examples of faults can be a link shutdown (Is there sufficient handshake to detect that the system must use an alternate path), memory value fault, and delayed sensor.

## Summary

System-level modeling and simulation for timing, power and functional analysis is a design validation and functional verification platform. This platform can be used early in the design process and, complementary to physical efforts such as a prototype board and algorithmic simulation such as MatLab. Analysis with VisualSim ensures the specification exceeds the requirement and operates at a satisfactory level in failures and irregular operations. The modeling process enables the designers to consider conditions that cannot be replicated in the physical prototype and other physically improbable failure situations. Using this knowledge, designers can develop contingency plans and, provide redundant processing, establish handshakes, create watchdogs to detect failure conditions. This will ensure a longer and safer product life.