Building TESSA

Some simulations using the Testbed Environment for Space Situational Awareness (TESSA) are based on techniques widely used at the Laboratory. For example, hydrodynamic simulations of the February 10, 2009, collision near the North Pole between a defunct Russian satellite and a privately owned American communications satellite show processes that occur continuously over time. The simulations mathematically break the collision, or intercept, into a grid and calculate all of the interactions that occur over the 100-millisecond time span of the collision and breakup.

Other aspects of TESSA simulations are more unique. Modeling the activity of radar systems and telescopes that track objects orbiting Earth requires a completely different simulation methodology. A telescope may pan the sky keeping stars in a fixed position. Satellites and other orbiting objects move in and out of the field of view, creating streaks across the sky. Radar is often programmed to jump around the sky, collecting information from various areas in quick succession. “To simulate the tracking of orbiting objects, we are examining discrete changes in state, not a continuous process,” says Livermore’s David Jefferson, who designed the TESSA framework. “Discrete event simulation is primarily concerned with discontinuities in a system’s behavior rather than the continuous parts.” Examples of other situations that require discrete event simulation are missile defense, national infrastructure, computer networks, particle systems, and air traffic control.

In the 1980s and 1990s, long before he arrived at the Laboratory, Jefferson worked with other experts around the country to develop methods for parallel discrete event simulation (PDES). The TESSA PDES architecture is based on two Livermore programs, Babel and Co-op. Babel earned a 2006 R&D 100 Award for its flexibility in communicating among programs written in different programming languages. High-performance applications in different languages can interoperate, allowing them to pass scientific data seamlessly and efficiently from one another. Co-op was built upon Babel and is a tool that allows parallel components to run different codes at the same time. The Co-op style of parallelism is described as “multiple programs, multiple data,” in contrast to “single program, multiple data,” the usual style of parallelism for scientific computations and simulations. A single processor may be able to simulate all of the data from a radar device, but multiple processors are needed to simulate what a telescope sees, and TESSA accommodates that difference.

In a continuum simulation, all parallel processes need to be synchronized in time. In PDES, however, the processors are not all handling data from the same moment in simulation time. “The big challenge with PDES is maintaining enough synchronization that all processors are used efficiently,” says Jefferson. “The processors handling data farther ahead in time cannot interact with those that are behind. We have to maintain causal relationships, which are always directed forward in time. Livermore is good at big simulations on big computers. TESSA is a striking new example.”