A Simulation and Modeling Framework for Space Situational Awareness

Scot S. Olivier
Lawrence Livermore National Laboratory

ABSTRACT

This paper describes the development and initial demonstration of a new, integrated modeling and simulation framework, encompassing the space situational awareness enterprise, for quantitatively assessing the benefit of specific sensor systems, technologies and data analysis techniques. The framework is based on a flexible, scalable architecture to enable efficient, physics-based simulation of the current SSA enterprise, and to accommodate future advancements in SSA systems. In particular, the code is designed to take advantage of massively parallel computer systems available, for example, at Lawrence Livermore National Laboratory. The details of the modeling and simulation framework are described, including hydrodynamic models of satellite intercept and debris generation, orbital propagation algorithms, radar cross section calculations, optical brightness calculations, radar system models, optical system models, object detection algorithms, orbit determination algorithms, simulation analysis and visualization tools. Future work will involve collaborative incorporation of specific Space Surveillance Network models, along with evaluation of currently unexploited sensors and prospective new sensors. There is also the possibility of providing this simulation and modeling framework to the community through a service oriented architecture.

1. INTRODUCTION

Lawrence Livermore National Laboratory (LLNL) applies forefront science and technology to anticipate, innovate and deliver responsive solutions to complex global security needs. LLNL is currently working with other National Labs (LANL, SNL, AFRL) on research and development for improving space situational awareness. In particular, LLNL is currently developing a comprehensive framework for modeling, simulation and visualization of space situational awareness. This framework, called the Test-bed Environment for Space Situational Awareness (TESSA), employs a parallel, discrete event simulation architecture utilizing high performance computing infrastructure. This provides a framework capable of (1) simulating the orbits of hundreds of thousands of objects, (2) physics-based performance modeling of multiple sensors, (3) scalable visualization in an interactive environment.

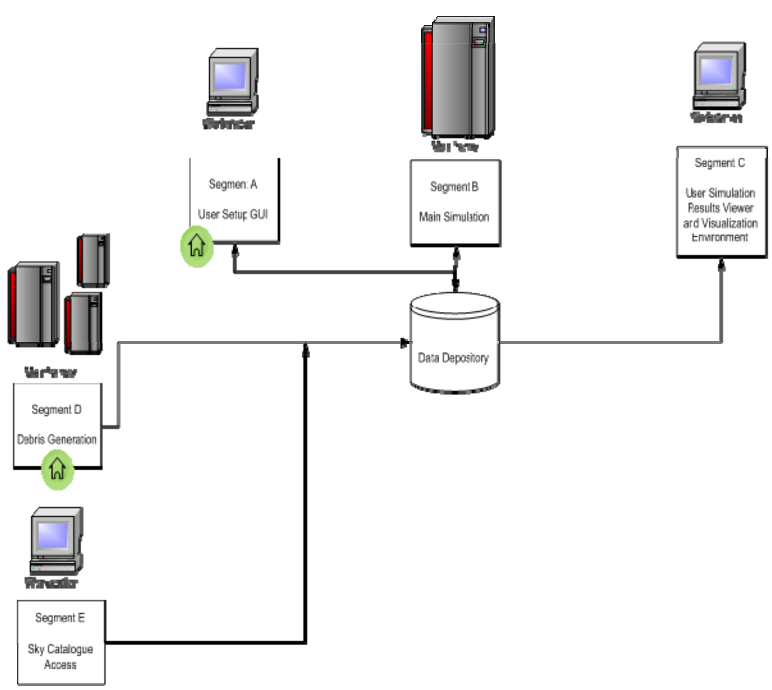

Fig. 1. Test-bed Environment for Space Situational Awareness: organization across multiple computer platforms

TESSA employs a modular simulation approach. Modules include hydrocode-based intercept debris generation, detailed orbital modeling, radar sensor models, optical sensor models, orbit determination, simulation analysis, and visualization.

The organization of TESSA across multiple computer platforms is shown in Fig. 1. The intercept debris generation requires of order 1000 CPU days, and is performed on a high performance computing system, e.g., a system with ~10,000 CPU's. Intercept simulations of satellites of interest may be pre-computed and the results stored. Effects due to changes in the intercept parameters (e.g., relative velocity, position and angle of impact, etc.) can often be modeled from pre-computed results using interpolation, or empirical scaling relations. The sky catalog, including both orbital and astronomical objects, is accessed from a standard desk-top machine. The user set-up GUI can also be run on a standard desk-top machine and enables the user to select relevant simulation parameters. The main simulation, composed of orbital models, sensor models and analysis algorithms (including orbit determination), also runs on a high performance computing platform since the physics-based sensor performance modeling can take of order 1 CPU day per day of simulated observation – e.g., a 30 day simulation with 30 sensors can take of order 1000 CPU days. Finally, the simulation visualization environment is run on a high-performance graphics workstation.

2. TESSA ARCHITECTURE

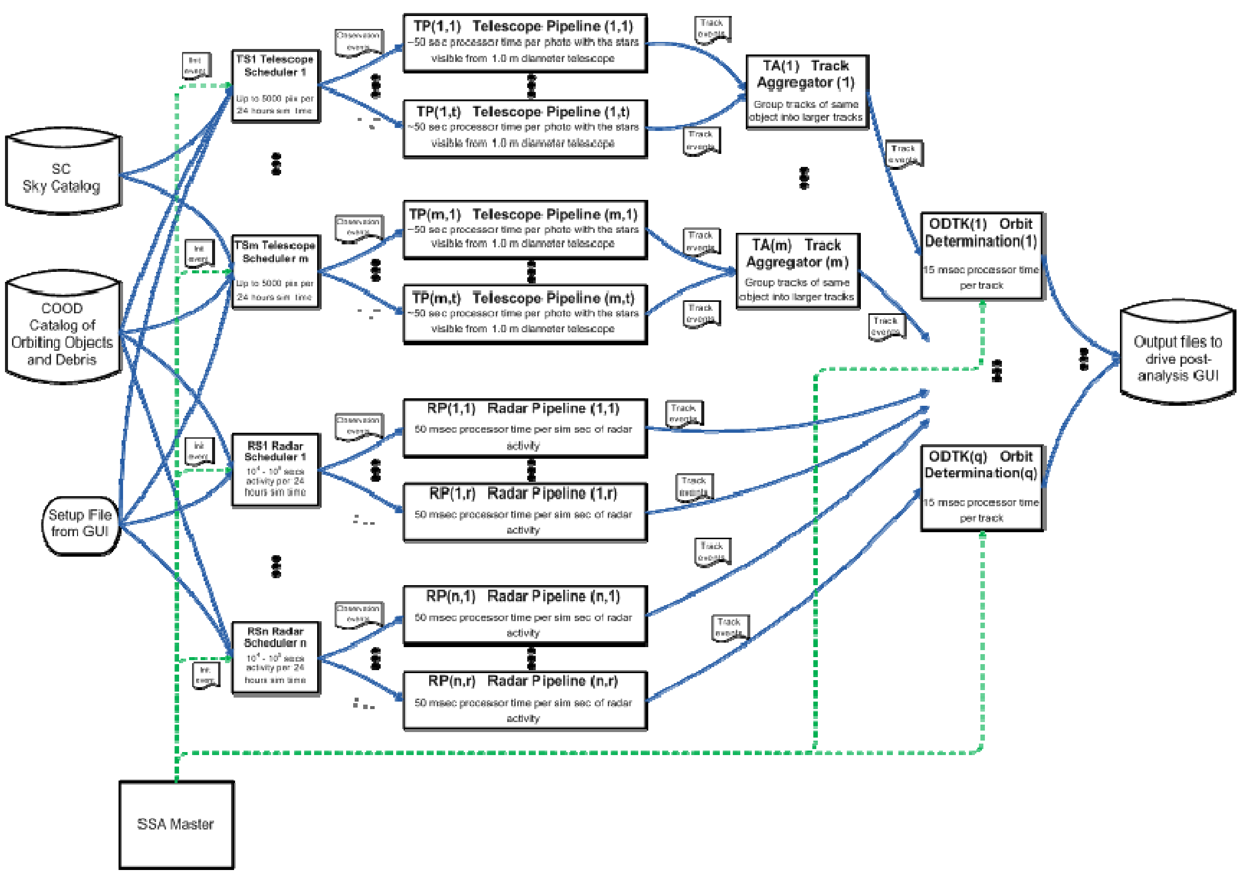

Fig. 2. TESSA architecture

Fig, 2 shows the TESSA architecture. The SSA Master Process launches all of the others, passes setup parameters to them, and terminates when all of the other processes terminates. The Telescope Scheduler decides when each telescope will take picture and on what patches of sky. The Radar Scheduler decides when each radar will take observation, in what patch of sky, and for how many seconds. The Telescope Pipeline creates simulated sky images, then extracts the tracks of orbiting objects from them and finally reduces the tracks to a few observations each. The

Radar Pipeline creates simulated radar observations. The Track Aggregator processes takes observations from several different tracks on several different photos of the same object from the same telescope and combines them into a single track long enough for orbit determination. The ODTK process determines orbits from the track data that it gets from the Telescope Pipelines (by way of the Aggregators) or the Radar PipeLines.

3. INTERCEPT DEBRIS GENERATION

Simulation of intercept debris generation can be used to establish accurate debris fragment size, shape, mass, and velocity for orbital propagation and coincidence studies. Modeling of hypervelocity impact and resulting debris generation is performed using PARADYNE, a parallel version of DYNA 3-D with an advanced, anisotropic fragmentation model. This code has been validated using light gas gun and sled track ground truth data. An example of the output from this code is shown in Fig. 3.

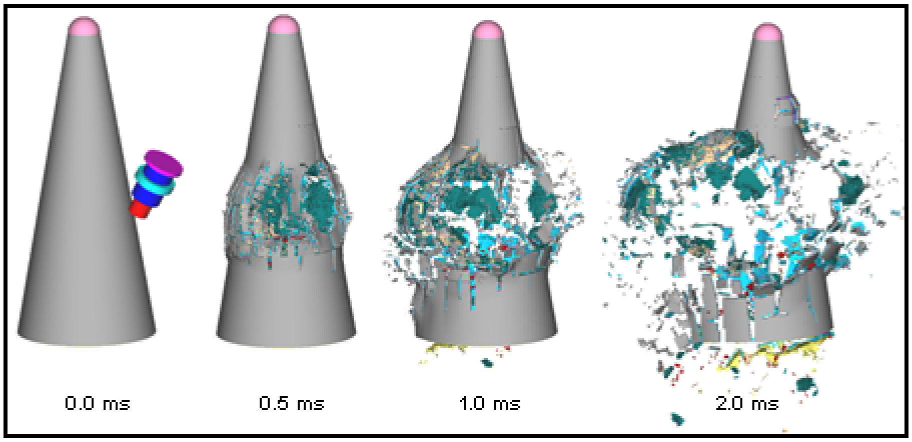

Fig. 3. Hydrocode intercept simulation

Once the hydrocode intercept simulation has completed, it is necessary to analyze the results to determine the detailed properties of the debris field. This process involves processing the results of the hydrocode intercept simulation to determine the chunks, i.e., blocks of elements connected by at least one face, and the free nodes, i.e., points not part of any element. Then, for each chunk and free node, the location and velocity of center of mass, mass, volume, and material content are computed. Finally the outer surface of the chunks is extracted, and the free nodes are converted to cuboids for plotting and estimation of radar cross sections and optical brightness.

4. ORBITAL MODELING

Orbital propagation is computed using a standard propagator, SGP4 (November 2007 update), or using a force model when higher accuracy is needed. Force models use either the JGM-3 or EGM-96 geopotential model and include solid body tides without resonance terms, sun and moon third body perturbations, along with Venus, Mars, and Jupiter third body perturbations. Different atmospheric models may be used, including Harris-Priester, NRL-MSISE-90, NRL-MSISE-2000, GOST 84, GOST 2004, Jacchia-Roberts 1971. The 1980 IAU 108 term Earth nutation model is used, as well as the IAU Earth precession model, and the ICRF inertial coordinates (J2000). Additional effects include solar radiation pressure, calculated using cylindrical or dual cone umbra and penumbra.

5. RADAR SYSTEM MODELS

The radar detections are performed using detailed models of radar cross sections, and radar system parameters. The initial implementation of the radar simulation employs a simple model of a “phased array” scanning, broad azimuth and elevation scanning with 1 second framing. The simulation of the radar pulse results in radar “returns” at 1 second framing intervals for the radar scan volume. Satellites and debris locations are computed, spatially-filtered and summed within the radar cross range and range bins to provide a single target response. The output from the radar simulation consists of the object azimuth, elevation, range, and signal level (summed radar cross section). The radar cross section (RCS) is computed for multiple object orientations, using either physical optics formulae or the LLNL Eiger code (a frequency domain boundary element simulation code), and then a random RCS is selected from the computed histogram and assigned to each object. Object tracks are formed from multiple observations for use in the orbit determination.

6. OPTICAL SYSTEM MODELS

The optical detections are performed using a standard astronomical image simulation code, SkyMaker, which includes detailed models of optical brightness and telescope system parameters. This results in realistic astronomical images. Scattered sunlight, moonlight, sky background and the star catalog, USNO-B1.0, are added as inputs. The code also simulates low order telescope aberrations, atmospheric turbulence, CCD blooming, aureole scattering, and tracking errors. A simulated optical image is shown in Fig. 4.

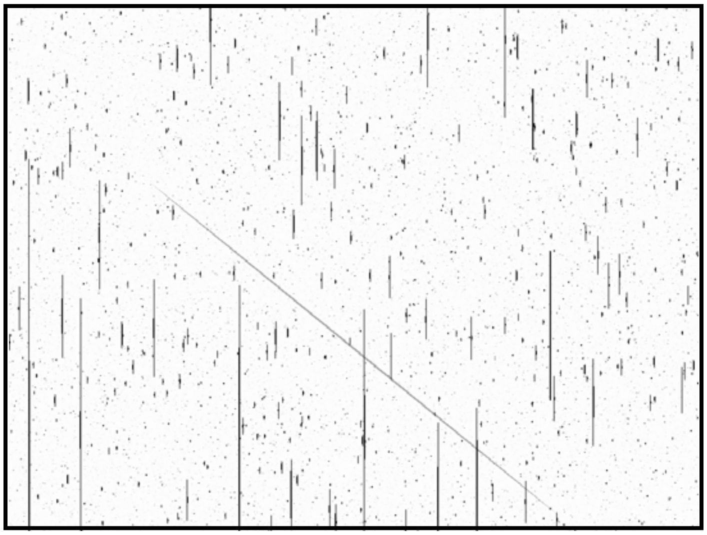

Fig. 4. Simulated optical image

The image analysis is performed using another standard astronomical code, Source Extractor. This code is used in the simulation to analyze the images produced by SkyMaker. The code builds a model of the sky background from the image, then subtracts the background, filters and thresholds image. Object detections are then cleaned, photometrically quantified, classified, and the results written to an output catalog. Fig. 5 shows an example of an image processed with Source Extractor.

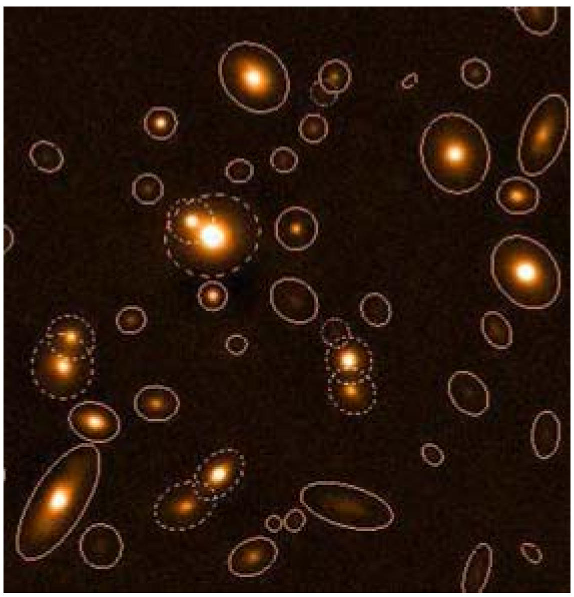

Fig. 5. Example of processed optical image

7. ORBIT DETERMINATION

Orbit determination is performed using the following steps:

1) First orbit determination
2) Orbit refinement using batch least squares
3) Follow orbit evolution due to unmodeled forces with the Extended Kalman Filter

Multiple methods for first orbit determination are available and may be utilized depending on the type of sensor data being simulated and the required accuracy. First order orbit determination from line of sight data may be computed using:

1) Laplace's method using three finely spaced lines of sight at known times,
   (No initial guesses are required; positive real roots of an eighth degree polynomial give possible values for r2; straightforward to identify which real root is the correct one when there is more than one; works well for orbital platforms as well as for earth-based sites – robust.)
2) Gauss's method using three widely spaced lines of sight at known times,
3) Escobal Double R method using three widely spaced lines of sight at known times,
4) Gooding's method using using three widely spaced lines of sight at known times.

First order orbit determination from position data may be computed using:

1) Lambert's method using two positions,
   (Two positions, an elapsed time, and the number of completed revolutions are required – very robust.)
2) Gibb's method using three widely spaced positions,
   (Three points on a branch of a conic and the focus uniquely specify the conic; time information is not used at all – very robust.)
3) Herrick-Gibbs method using three finely spaced positions
   (A Taylor series expansion in the line of sight direction as a function of time is done – very robust.).

8. SIMULATION ANALYSIS

Automated post-processing of simulation results can be performed. For instance, a tool has been developed for assessing the efficiency of a simulated space surveillance network (or any collection of sensors) in determining orbits for new objects (due to satellite break-up or maneuvers). This analysis tool can compute the efficiency as a function of various parameters: time, mass, area, orbital energy, etc., or combinations thereof. An example of this analysis is shown in Fig. 6.

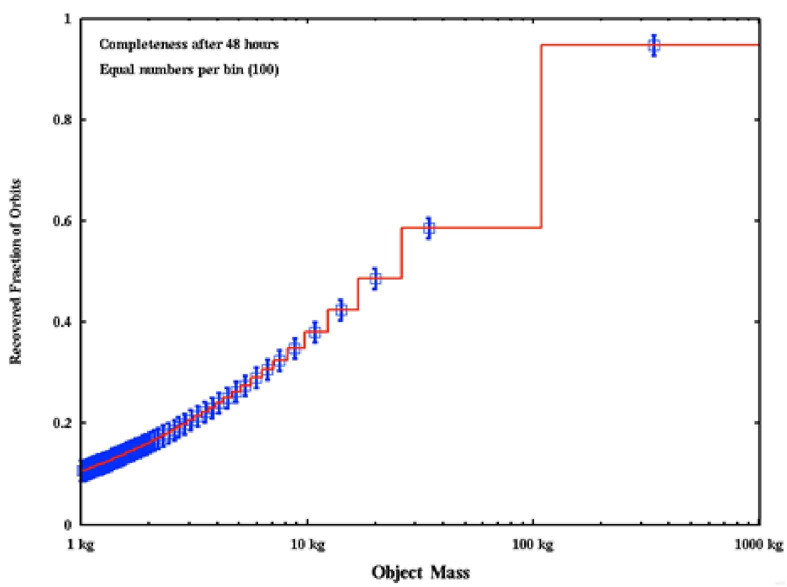

Fig. 6. Example of simulation analysis

9. VISUALIZATION

An effective visualization system for SSA enables the user to:

1) Browse and navigate large amounts of data
2) Provide real-time rendering of the simulation data
3) Allow interactive queries and simulation feedbacks

Our SSA visualization system incorporates three main components: interactive analysis and feedback, progressive real-time rendering, advanced data storage layout.

An example of our visualization output is shown in Fig. 7.

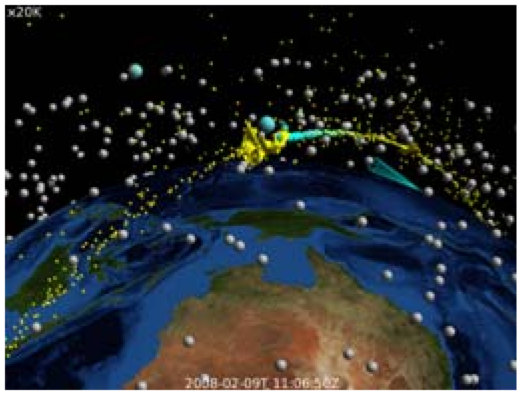

Fig. 7. Visualization example

10. STATUS AND FUTURE WORK

An initial implementation of the TESSA modeling and simulation framework, incorporating all the major elements described above has been completed. Furture work will involve incorporating and validating specific SSN sensor and scheduling models (in collaboration with ESC, AFSPC/A9, AFRL/HPCC, etc.), as well as implementing specific new sensor models and analyzing scenario-dependent system performance enhancement. We have also begun to explore how to make the framework available to the community through a service oriented architecture.

11. ACKNOWLEDGEMENTS

This work was performed by the LLNL SSA development team: Alex Pertica, David Jefferson, Thomas Edmunds, James Gansemer, Hal Brandt, Kem Cook, Kevin Baker, Sergei Nikolaev, Willem De Vries, Ron Kane, Dan White, Ben Fasenfest, Nicholas Matyas, Keo Springer, Joanne Levatin, Ming Jiang, Peer-Timo Bremer, Ross Gaunt, Don Phillion, Ken Kim, Celeste Matarazzo, Valerio Pascucci, Mark Stowell.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.