OPEN SOURCE SOFTWARE SUITE FOR SPACE SITUATIONAL AWARENESS AND SPACE OBJECT CATALOG WORK

EUROPEAN SPACE ASTRONOMY CENTRE (ESA/ESAC), MADRID, SPAIN
3 – 6 MAY 2010

Paul J. Cefola [(1)], Brian Weeden [(2)], Creon Levit [(3)]

[(1)] Consultant, 59 Harness Lane, Sudbury Massachusetts 01776, USA, (also Adjunct Faculty, University at Buffalo, SUNY), Email: paulcefo@buffalo.edu

[(2)] Technical Advisor on Space Security and Sustainability Issues, Secure World Foundation, 5610 Place Bayard, Brossard, Quebec, J4Z 2A5, Canada, Email: bweeden@swfound.org

[(3)] Chief Scientist for Projects and Programs, NASA Ames Research Center, Moffet Field, California 94035, USA, Email: creon.levit@nasa.gov

ABSTRACT

The accuracy of the orbital data products used for space situational awareness is affected by the evolution of the sensors collecting the data, the knowledge and control of the errors in the sensor network, the knowledge of the space environment, the available computing resources (both hardware and software), and the number of space objects to be monitored. While the number of objects in orbit has grown significantly over the last three decades, the quality and quantity of the orbital data products, particularly those available to non-government space operators has not kept pace. Further, operational analysis of key issues is still in flux. The Iridium/Cosmos collision event in 2009 demonstrated that there is a lack of data and tools available to all space actors needed to avoid major accidents. Additionally, the event demonstrated that the publicly available Two Line Element (TLE) sets are not sufficiently accurate to detect and prevent such incidents.

Open source software is a relatively new trend in software development that rests on the principles of open collaboration. Unlike closed source software, the source code behind the software is publicly released and anyone is free to modify it. These modifications can become a completely new project or be integrated into the mainline development. Critically for the space surveillance application, software developed using open source software methods provides greater transparency, knowledge of what's inside the "black box", flexibility, can reduce development costs, and can be used globally with fewer restrictions in regard to export controls and intellectual property restrictions.

The focus of this paper is on the software aspects of moving the current legacy space situational awareness capabilities forward via an open source paradigm, so that all spacecraft operators have access to the basic tools needed to operate safely and efficiently in space.

1. INTRODUCTION

The fundamental requirement of space situational awareness (SSA) is to provide actionable knowledge about events and activities in Earth orbit. A key component of SSA is space surveillance -- determining the present position of space objects and the ability to predict their future orbital paths. Related requirements are the detection of new space objects, the detection of spacecraft maneuvres, and the prediction of when one space object may interfere with another space object. Such interference may be physical in nature such as the February 2009 collision between the Iridium 33 and Cosmos 2251 or electromagnetic in nature such as the Galaxy 15 anomaly[1].

All of these requirements require space object ephemeris data; an Ephemeris is a table of predicted position and velocity at a sequence of times, usually at equal time intervals. The ephemeris data is generated by fitting mathematical models to tracking data. The tracking data includes data from radar and optical sensors. The radar sensors include phased array radars, dish radars, and fences. The orbit fitting process makes use of the residual between the actual measurement at time t and the computed measurement[2] at time $t+\delta t$ where the quantity δt is the timing bias.

[1] For a more detailed description of the Galaxy 15 situation, see "Dealing With Galaxy 15: Zombiesats and On-orbit Servicing", *The Space Review*, 24 May 2010, online at http://thespacereview.com/article1634/1

[2] The computed measurements require an a priori estimate of the state vector.

Early in the space age, national governments were the primary developers and operators of systems for space surveillance. In the U.S., the initial systems were the Navy SPASUR (Space Surveillance) System and the Air Force Interim National Space Surveillance Control Center (INSSCC).

From an astrodynamic algorithmic point of view, the US systems rapidly converged on a specification including analytical orbit propagators based on the Brouwer theory and a numerical orbit propagator based on an arbitrary geopotential model. The orbit determination process was based on the batch weighted least squares method adapted to nonlinear dynamical models.

U.S. SSA systems also made the assumption of centralized processing: all the observations were brought to a data processing facility in Colorado Springs, CO. The Navy data processing facility in Dahlgren, VA, processed the raw observations from the Navy Fence (now the Air Force Space Surveillance Fence) and functioned as backup for the space surveillance operations in Colorado Springs.

Over the time period from 1957 to present, two major trends have impacted SSA. The first is the continuously increasing number of objects in space (Figs. 1 & 2). The phrase 'trackable objects' (Fig. 1) refers to 10 cm objects for LEO and 1 m objects for GEO.

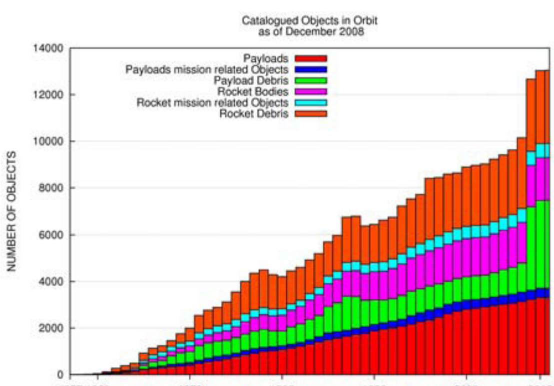

Figure 1. Evolution of the number of trackable, on-orbit objects [1]

Fig. 2 gives a projection of the number of currently trackable space objects two centuries into the future under a pessimistic assumption with respect to post-mission disposal.

In fact, if we think about object sizes down to 2 cm, the number of objects currently in space is on the order of hundreds of thousands.

The second major trend is the evolution of performance in scientific computing on the large scale. The life span of modern computing overlaps the space age.

Fig. 3 gives the performance of various computers of the last five decades that could have been called the 'supercomputers' of their time. On the average, there is an increase in performance of two orders of magnitude every decade.

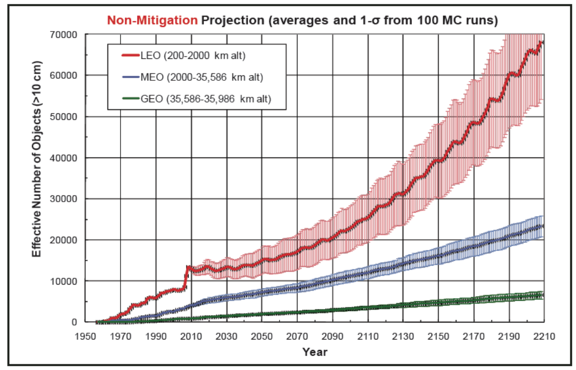

Figure 2. Projection of the growth of > 10 cm resident space objects including Fengyun-1C ASAT and Iridium/Cosmos collisions if post-mission disposal measures are not implemented [2]

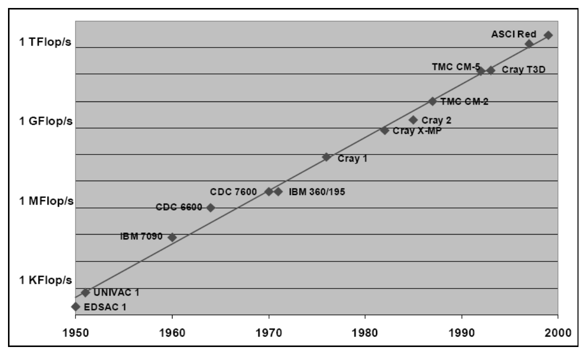

Figure 3. Moore's Law and peak performance of various computers over time [3]

In addition to improvements in computer hardware improvements, there has been continuous improvement in the software development environment (particularly configuration management tools) and the operating systems for large scale projects.

The availability of improved computer technology together with the increase in the number of objects in space and evolution in the desired accuracy of the space catalogue has led to the almost continuous development of new U.S. SSA systems, with little actual delivery of capabilities to the end user. The current developmental system is the JSpOC Mission

System (JMS) being developed by the U.S. Air Force Space Command for the Joint Space Operations Center (JSpOC) located at Vandenberg AFB, CA. The JMS is slated to replace the current SPADOC 4C and CAVENet systems.

Development of SSA systems by the other major space-faring nations, particularly Russia and the FSU, also has resulted in significant capability. However, different levels of sensor, analytical, computer, and communications capabilities resulted in different SSA system architectures [4] and algorithms [5].

There have been limited efforts to compare the US and Russia catalogs [6][7].

While the current US developmental system is based on improved computer technology, the algorithmic content of the system as of early 2009 was the same General Perturbations (GP4) analytical orbit propagator, numerical integration-based Special Perturbations orbit propagator, and batch least squares orbit determination process originally put in use 35-years earlier (1975).

The following list gives capabilities that are not addressed in the current operational system used by the JSpOC:

- Observation compression concepts are not available for either radar or optical sensors
- Fast and accurate orbit propagator concepts are not available
- Fast and accurate state transition matrix concepts are not available
- Kalman Filter-based orbit estimation concepts are not available
- Kalman Filter-based sensor calibration processes for are not available
- Realistic process noise and measurement error models are not employed
- The orbit uncertainty as represented by and propagated by the orbit determination systems is not well understood
- The operational processes developed by the AF Space Command for real time tracking of the atmospheric density variations are limited and narrow in scope
- There is no process for re-acquiring a significant portion of the catalog, as would be required in the event of a major geo-magnetic storm (such as 1989)
- There is no mathematically 'strong' theory for the general concept of catalog maintenance including both the detection and tracking processes, either for LEO, GEO, or HEO
- There is no concept for taking advantage of frameworks that can be massively parallelized on distributed memory clusters.
- There is no web services based architecture for SSA
- There is no capability for utilizing the very large datasets that will result from larger catalogs and improved sensors[3]
- There is little cooperative, positive relationship between the USG SSA community and the broader astrodynamics research community

The current paper is part of a larger strategy to address the SSA problem via an Open Source Software paradigm. In [8], Weeden and Cefola address "Mathematical Algorithms for Space Situational Awareness: History and Future Development." In [9], Weeden, Cefola, and Sankaran will address "Global Space Situational Awareness Sensors."

The roadmap of the current paper is as follows. In Section 2, we provide an overview of the problem of adapting SSA tools to a modern distributed computing infrastructure. In Section 3, we consider rewriting the SSA tools in a language platform employing object-oriented and component technologies such as C++/CORBA. In Section 4, we consider the more moderate approach of non-invasive encapsulation of legacy binaries. The configuration issues with legacy codes are shifted from hand-edited files and scripts to automatically generated GUIs. In Section 5, we give a specific plan for creating a Web 2.0 architecture for SSA.

2. APPROACHES FOR DEALING WITH SOFTWARE LEGACY

Computer programs for the determination of the orbits of artificial satellites and space debris objects require models from several disciplines:

- Nonlinear estimation
- Measurement modeling
- Force modeling

[3] We assume that the SSN will be improved both qualitatively and quantitatively.

- Numerical analysis

Such programs are generally complicated with several hundreds or thousands of modules. These programs tend to be written in Fortran 77 because that was the language of choice when the development of these programs started. Table 1[10] gives a list of OD programs with active development communities and recent enhancements. These programs tend to have very long lifetimes. Major development efforts were accomplished in the 60s and 70s when the state of art in software development environments was not advanced.

Table 1. Organization-specific Orbit Propagator and Determination Programs and Applications [10]

Organization	Software program	Primary application
Aerospace Corporation/USAF	TRACE	Operational OD evaluation and covariance analysis www.aero.org/publications/crosslink/summer2002/04.html
Analytical Graphics Inc.	STK/HPOP	Integrated graphics and numerical processing www.agi.com/products/desktopApp/odtk
Charles Stark Draper Laboratory	DSST DGTDS	Precision semianalytical OD technique www.csdl.org POD
APL	OIP/ODP	Transit Doppler post-processing OD used in the 1960s through the 1980s
MICROCOSM	MICROCOSM	Commercial software OD package of the NASA GEODYN program www.vmsi_microcosm.com
MIT/LL	DYNAMO	POD, specifically for HEO and GEO satellites www.ll.mit.edu
NASA/GSFC	GTDS RTOD	Operational OD for LEO, MEO, and GEO orbits (TDRSS) and lunar and interplanetary orbits fdab.gsfc.nasa.gov/live/Home/Tools_Nav_GTDS.html Precision real-time OD for onboard spacecraft using Kalman filtering nctn.oact.hg.nasa.gov/ft-tech-GEONS.html
NASA/GSFC	GEODYN II	POD for geodesy and geophysics bowie.gsfc.nasa.gov/697/POD/POD.html
NASA/JPL	MIRAGE	Multiple satellite OD using GPS
NASA/JPL	DPTRAJ	Interplanetary OD
NASA/JPL	GIPSY/OASIS II (GOA)	POD of satellites using GPS, SLR, and DORIS observations gipsy.jpl.nasa/orms/goa
Navy/NSWC	OMNIS/EPICA	GPS precision orbits earth-info.nga.mil/GanG/sathtml/gpsdoc2006_11a.html
Navy/NSWC	PPT3[a]	Surveillance and space debris tracking and propagation
Navy/NSWC	Special-K	Operational numerical OD program
Navy/NRL	OCEANS	Orbit studies, covariance analyses, and GPS orbits www.nrl.navy.mil
SAO	DOI	Used in the early 1960s for OD of Baker-Nunn camera data and development of standard Earth gravity models
USAF/SPACECOM	MCS	GPS operational orbits
USAF/SPACECOM	SGP4[a]	Surveillance and space debris tracking and propagation
USAF/SPACECOM	SPADOC/ SPECTR	Operational numerical OD program used by Shreiver and Kirkland AFBs
USAF/SPACECOM	ASW	Workstation numerical OD program
University of Texas	UTOPIA, MSODP	Precision orbits using GPS, SLR, and DORIS observations; TRANET, OPNET, altimetry www.csr.utexas.edu

We note that the SPADOC and CAVENet (ASW) systems are included in Table 1.

The systems listed in Table 1 taken together are the starting point for the Open Source Software for SSA project.

When we consider transformation of legacy computer programs with significant scientific computing requirements to a modern distributed computing infrastructure, there are two major options: ***Migration*** and ***Encapsulation*** [11].

Migration refers to rewriting all the applications in a language platform employing object-oriented and component technologies such as C++/CORBA [12]. The migration approach is costly in terms of programming effort and accounting for all the evolutionary work done to date.

Encapsulation is an alternative to migration. A non-invasive approach to encapsulation is to employ the legacy binaries in predefined (but configurable) workflows with all the data exchange between binaries continuing to take place through file I/O [11]. This way of working with legacy codes reduces to devising an extensible encapsulation of the software components that treats them as black boxes with a set of inputs/outputs and a set of valid types and ranges of compile-time and run-time parameters [11].

3. MIGRATION

Our plan is to adapt key SSA algorithms to a modern distributed computing environment by rewriting the applications in an object-oriented language platform such as C++ [12].

The SSA algorithms to be considered include:

- Observation compression
- Orbit Propagators
- State Transition matrix
- Improved nonlinear Kalman Filters
- Realistic process noise
- Atmosphere density variations
- Observation data association
- Observation data simulation

The objective of this effort is object-oriented programs written in a modern language such as C++. We intend to include a variety of algorithmic approaches for each of these functions. For example, for the orbit propagator, we plan to include:

- Numerical Integration (Special Perturbations) with high degree and order geopotential and modern atmosphere density models
- DSST (Semi-analytical Satellite Theory) [13] with high degree and order geopotential and modern atmosphere density models
- Brouwer-Lyddane (General Perturbations) [14]
- NORAD GP (SGP, SGP4, SGP8) [15] (with tesseral m-daily option)
- NORAD HANDE [16] (with tesseral m-daily option)
- NAVSPASUR PPT [17] (with tesseral m-daily option)
- Russian A and AP (GP) [5]
- Russian NA Numerical-Analytical with improved accuracy features as in the AP theory [5]

For orbit determination, we plan to include:

- Batch (mean element estimation as an option)
- Extended Kalman Filter (mean element estimation as an option) [18]
- Modern Filters (mean element estimation as an option) [19]

The intent is to allow a wide range of orbit determination comparison studies.

To clarify the issues in the migration project, we are undertaking a demonstration task:

- Migration of the Standalone Draper Semi-analytical Satellite Theory (DSST) from Fortran 77 to Object-Oriented C++

As part of this effort, we plan to study the application of object-oriented design principles in the Generalized Mission Analysis Tool (GMAT) [20] program.

The plan is to accomplish this demonstration task using a modern Graphical Server with multiple CPUs and graphical CPUs. We will be able to study the parallelization of the semi-analytical theory using OpenMP [21] for CPUs and Cuda for GPUs [22].

4. ENCAPSULATION

The orbit propagator and orbit determination programs listed in Table 1 represent a tremendous investment in resources for the respective organizations. These programs tend to be extremely long-lived. Typically, the development started as early as 1970 on mainframe computers and these programs have been ported to multiple additional generations of computers:

- 32-bit minicomputers (such as the VAX)
- Workstations
- Micro-computers

Many of these programs were coded in Fortran and were revised by different engineering teams. As time progressed, difficulty often arose when an attempt was made to port the software, make significant improvements, or add new features.

This was particularly frustrating with the availability of significant new software technology (such as object-oriented design) and the new hardware technology (such as multi-core, multi-thread machines).

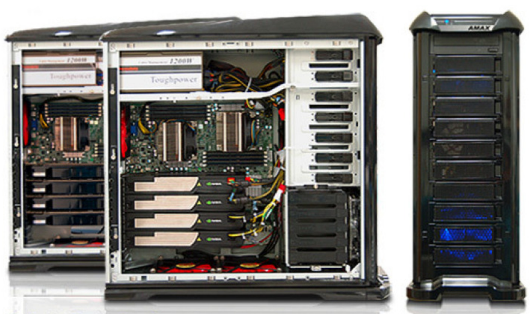

Figure 4. Personal Supercomputer Design (2010)

Such considerations lead directly to the consideration of **migration** and **encapsulation**. We have discussed migration previously.

The software legacy issue is not unique to the orbit propagation and orbit determination communities. The field of interdisciplinary ocean prediction systems is another example of a field that experiences the software legacy issue [11].

A non-invasive approach to encapsulation is to keep employing the legacy binaries in predefined (but configurable) workflows with all data exchange between binaries continuing to take place through file I/O [11]. The binaries configuration however is shifted from hand-edited files and scripts to automatically generated GUIs. This way of working with legacy codes reduces to devising an extensible encapsulation of the software components (as binaries) that treats them as black boxes with a set of inputs/outputs and a set of valid types and ranges of compile-time and run-time parameters. The advent of XML provides a standards-based way to accomplish this. XML describes data through the use of custom tags thus eliminating the need to conform to a specific programming structure and offering the possibility to integrate legacy software with new technology.

The ocean prediction community has developed software tools for addressing the non-invasive encapsulation problem [23].

We propose to extend non-invasive encapsulation techniques to significant SSA applications that will not be rewritten.

To clarify the issues in encapsulation, we are undertaking a demonstration task:

- Non-invasive encapsulation of the Linux GTDS R&D Orbit Determination system using LCML and LEGEND [23]

The 2010 version of Linux GTDS with expanded sensor modeling will be employed. The source code, makefile, and keyword descriptions will be inputs to the encapsulation process. We hope to undertake this demonstration using a modern Graphical Server with multiple CPUs and graphical CPUs.

5. WEB 2.0 ARCHITECTURE FOR SSA

Our goal is make international Web services-based tools for Space Situational Awareness and Space Traffic Management that are freely available to all satellite operators and others who need to operate safely and efficiently in space. Thus a user sitting anywhere in the world would be able to operate on his own 'data' via the web. 'Data' might range from a nominal orbit or constellation for which the user was trying to understand the long term motion or to the raw observations (range, azimuth, elevation, range-rate, right ascension, and declination) at multiple times which the user was trying to filter to create an element set. The user would be able to operate on his data with algorithms and software that were transparent to him.

A more specific expression of the goal is to create a Web 2.0 architecture for a SSA service based on the human-provided services (HPS) paradigm [24].

To clarify the issues in this part of the project, we propose a demonstration task to create an initial design of the Web services-based architecture for SSA

6. CONCLUSIONS

Overall, we have three major technical goals for our project:

1. To adapt key algorithms to a modern distributed computing environment by

rewriting the applications in a language platform object-oriented and component technologies such as C++/CORBA [C]

2. To extend non-invasive encapsulation techniques to significant SSA applications that we don't rewrite
3. To create a Web 2.0 architecture for SSA based on the human-provided services (HPS) paradigm.

To develop this new SSA capability, we intend to develop an Open Source Software project following the insight of Karl Fogel [25].

To clarify the issues in the project, we have suggested three demonstration tasks:

7. ACKNOWLEDGEMENT

The authors would like to acknowledge our co-workers for their support during the technical work described in this paper. Particularly, we would like to acknowledge Mr. Bill Robertson (Technical Staff Emeritus at the Draper Laboratory) for many useful discussions of the early days of space surveillance (INSSCC, 496L, and SPADATS). We also wish to acknowledge several useful discussions with Dave Vallado (Center for Space Standards and Innovation, Colorado Springs).

The authors would like to acknowledge the representatives of the Space Surveillance community in both the USA and in Russia for supporting the USA-Russia Space Surveillance Workshops. These workshops have contributed greatly to the authors' knowledge of the international SSA.

The first author especially wishes to acknowledge very useful conversations with Prof. Andrey Nazarenko, Dr. Z. Khutorovsky, and Dr. V. S. Yurasov.

The first author would also like to acknowledge his long time colleagues, Dr. Ron Proulx and Dr. David W. Carter (at Draper Laboratory), Dr. Mark Slutsky (at Raytheon Company), and Mr. Zachary Folcik (at MIT LL).

The authors acknowledge the partial support of the Secure World Foundation, Superior, Colorado, for this research.

8. REFERENCES

1. Klinkrad, H., and Johnson, N. L., (2009). **Space Debris Environment Remediation Concepts**, presented at the Fifth European Conference on Space Debris, Darmstadt, Germany, March 2009.
2. **NASA Orbital Debris Program Office, Orbital Debris Quarterly News**, (2010) 14 (1), 1-2.
3. Dongarra, J. H., Meuer, H., Simon, H., and Strohmaier, E., (2000) **High Performance Computing Today**, http://icl.cs.utk.edu/publications/pub-papers/2000/hpc-today.pdf.
4. Khutorovsky, Z. N., (2004), **Techniques and Algorithms for determination of orbits of LEO satellites using measurements acquired during one penetration to the field of view of detection radar**, Moscow, 2004.
5. Boikov, V. F., Makhonin, G. N., Testov, A. V., Khutorovsky, Z. N., and Shogin, A. N., (2009), **Prediction Procedures Used in Satellite Catalog Maintenance**, Journal of Guidance, Control, and Dynamics, **32**(4), 1179-1199.
6. Aksenov, O., Andreyev, O., Dicky, V., Johnson, N., Tretyakov, Yu., Veniaminov, S., and Zavaliy, V., (2003), **Dynamics of the Principal Characteristics of the US/Russian Space Object Catalogs since 1995 to 2002**, In *Proceedings of the Fifth U.S.-Russian Space Surveillance Workshop*, 24-27 September 2003, Pulkovo, St. Petersburg, Russia, pp 21-33.
7. Dicky, V., Johnson, N., Tretyakov, Y., Veniaminov, S., Zavaliy, V., and Khutorovsky, Z., (2000), **Comparative Analysis of the Russian and United States Space Object Catalogs**, In *Proceedings of the Fourth U.S.-Russian Space Surveillance Workshop*, 23-27 October 2000, U.S. Naval Observatory, Washington, DC, USA, pp 389-398.
8. Weeden, B., and Cefola, P., (2010), **Mathematical Algorithms for Space Situational Awareness: History and Future Development**, presented at the *12th International Space Conference of Pacific-basin Societies (ISCOPS)*, 27-30 July 1010, Montreal, Quebec, Canada.
9. Weeden, B., Cefola, P., and Sankaran, J., (2010), **Global Space Situational Awareness Sensors**, to be presented at the *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 14-17 September 1010, Maui, Hawaii.
10. Vetter, J. R., (2007), **Fifty Years of Orbit Determination: Development of Modern Astrodynamics Methods**, Johns Hopkins APL Technical Digest, 27(3), 239-252.
11. Evangelinos, C., Lermusiaux, P. F. J., Geiger,

S. K., Chang, R. C., and Patrikalakis, N. M., (2006) **Web-enabled Configuration and Control of Legacy Codes: An Application to Ocean Modeling**, Ocean Modeling, 13, 197-220.

12. Serrano, M., Carver, D., Geiger, and Montes de Oca, C., (2002) **Reengineering Legacy Systems for Distributed Environments**, The Journal of Systems and Software, 64, 37-55.

13. Cefola, P., Phillion, D., and Kim, K. S., **Improving Access to the Semi-Analytical Satellite Theory**, AAS 09-341, presented at the AAS/AIAA Astrodynamic Specialist Conference, Pittsburgh, PA, August 2009.

14. Brouwer, D., **Solution of the Problem of Artificial Satellite Theory Without Drag**, Astronomical Journal, Vol. 64, No. 1274, pp 378-397 (November 1959) [work supported by a contract with the Air Force Cambridge Research Center, AF 19(604)4137].

15. Hoots, F. R., and Roehrich, R. L., **Models for Propagation of NORAD Element Sets**, Spacetrack Report No. 3, December 1980, Aerospace Defense Command, USAF.

16. Hoots, F. R., and France, R. G., **"An Analytic Satellite Theory Using Gravity and a Dynamic Atmosphere",** Celestial Mechanics, Vol. 40, No. 1, 1987.

17. **PPT2: The NAVSPASUR Model of Satellite Motion**, NAVSPASUR Report 92-01, July 1992 [available from Commander, NAVSPACECOM, 5280 Fourth Street, Dahlgren, VA 22448-5300; attention: Logistics and Information Systems Division, Mail Code N4/6].

18. Taylor, S. P., **Semi-analytical Satellite Theory and Sequential Estimation,** Master of Science Thesis, Department of Mechanical Engineering, MIT, September 1981 (CSDL-T-757).

19. Folcik, Z. J., **Orbit Determination Using Modern Filters/Smoothers and Continuous Thrust Modeling,** Master of Science Thesis, Department of Aeronautics and Astronautics, MIT, June 2008 (Lincoln Laboratory research project).

20. Conway, D. J., **General Mission Analysis Tool (GMAT) Architectural Specification** (Draft version), Thinking Systems, Inc., 2007

21. Mattson, T., and Meadows, L., **A "Hands-on" Introduction to OpenMP**, tutorial presented at the SC08 conference, November 2008, Austin, TX, accessed at http://www.openmp.org/mp-documents/omp-hands-on-SC08.pdf , July 30, 2010

22. Nickolls, J., and Dally, W. J., **The GPU Computing Era**, published by the IEEE Computer Society, 2010, accessed at http://www.computer.org/portal/c/document_library/get_file?uuid=6fefff2f-cf51-487b-b744-18ac6f3872ac&groupId=53319, July 30, 2010

23. Geiger, S. K., **Legacy Computing Markup Language (LCML) and LEGEND – LEGacy Encapsulation for Network Distribution,** Master of Science Thesis, Department of Ocean Engineering, MIT, June 2004.

24. Schall, Daniel, **Human Interactions in Mixed Systems – Architecture, Protocols, and Algorithms**, PhD Dissertation, Technical University Vienna, January 2009.

25. Fogel, Karl, **Producing Open Source Software: How to Run a Successful Free Software Project**, 2005, accessed at http://producingoss.com/, 28 July 2010.