STA, THE SPACE TRAJECTORY ANALYSIS PROJECT

Guillermo Ortega[(1)], Chris Laurel[(2)], Sven Erb[(1)], Carlos Lopez [(3)], Christof Bueskens [(4)], Erwin Mooij[(5)], Roland Klees[(6)], Fernando J. P. Lau[(6)], Paulo J. S. Gil[(6)], Joerg Fliege[(7)], Jose M. Giron-Sierra[(8)], Luis N. Vicente[(9)], Mark Burchell[(10)], Jan-J. Ruckmann[(11)], Matthias Gerdts[(12)], Michèle Lavagna[(13)], Ramón Martínez[(14)], Ricard González-Cinca[(15)], Bénédicte Escudier[(16)], Eusebio Valero[(17)], Cem Ozan Asma[(18)]

[(1)]European Space Agency, Keplerlaan 1 AG2200, Noordwijk, The Netherlands, Email:guillermo.ortega@esa.int [(2)]Periapsis Visual Software, Seattle, USA; [(3)]European Space Agency, ESAC, Spain; [(4)]University of Bremen, Germany; [(5)]Technical University of Delft, The Netherlands; [(6)]Instituto Superior Técnico, Lisbon, Portugal; [(7)]University of Southampton, United Kingdom; [(8)]Complutense University of Madrid, Spain; [(9)]Univeristy of Coimbra, Portugal; [(10)] University of Kent, United Kingdom; [(11)]University of Birmingham, United Kingdom; [(12)]University of Wuerzburg, Germany; [(13)]Politecnico di Milano, Italy; [(14)]Technical University of Madrid, Spain; [(15)]Technical University of Catalonia, Spain; [(16)]Institut Supérieur de L'Aéronautique et de l'Espace, Toulouse, France; [(17)]Technical School of Aeronautics, Madrid, Spain ; [(18)]Von Karman Institute, Brussels, Belgium

ABSTRACT

This article describes the objectives of the Space Trajectory Analysis (STA) project. The article also details the birth of STA, and its present configuration.

STA is a project to develop an open source astrodynamics software suite involving university science departments and space research institutions. It was initiated by ESA as internal activity in 2005 and now it involves 16 partners. The article explains the partnership into the STA Steering Board. The main purpose of the STA is to allow advanced simulation for the analysis of space trajectories in an open and free environment under the premises of innovation and reliability.

Further, the article explains that the STA development is open source and is based on the state of the art astrodynamics routines that are grouped into modules.

Finally, the article concludes about the benefits of the STA initiative: the STA project allows a strong link among applied mathematics, space engineering, and informatics disciplines by reinforcing the academic community with requirements and needs coming from real missions.

1. OBJECTIVES OF STA

The objectives of STA are threefold: first, to build an education environment in areas of astrodynamics, space sciences, mathematics, and informatics where teachers and students can interact towards Master, PhD, and postdoctoral degrees. Second, to promote the exchange of technical ideas, and raise knowledge and competence in these areas at University level. Last to provide a software suite for the purposes of research. In this sense, STA is conceived as test bed for the testing, verification and validation of new ideas, technologies, and techniques.

Conceived as an education and research tool to support the analysis phase of a space mission, STA is able to visualize a wide range of space trajectories. These include among others orbits around planets and moons, re-entry, descent and landing trajectories, rendezvous trajectories, etc (figure 1).

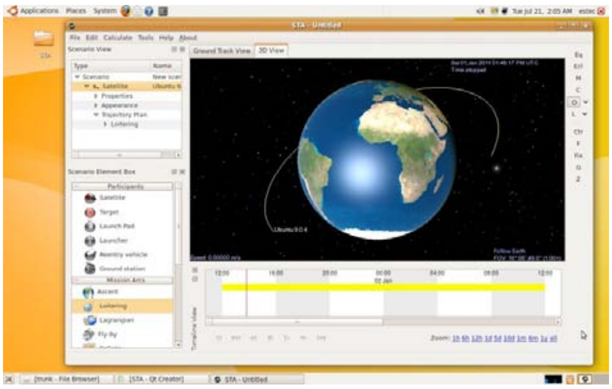

Figure 1. STA screenshot

The code of STA is open source and its modules are programmed using the C++ language. The different STA modules are designed, developed, tested and verified by the different Universities. Software integration and overall validation is performed by ESA. Students are chosen to work in STA modules as part of their Master or PhD thesis programs.

As part of their growing experience, the students involved in the development of the Space Trajectory Analysis project learn how to write documentation for a space project using European Cooperation on Space Standardization (ECSS) standards [R1]. In this scope, students learn how to test and verify the software modules they write and how to interact with ESA and partnering institutions in this process.

2. STA COMPONENTS and DEPENDENCIES

Thanks to the Qt development framework [R8], STA is an analysis environment that comes in three flavors:

Windows, Linux and MAC OS X (see figure 2). Identical functionality is provided in all operating systems supported. The widgets are operating system native and provide the user the same look and feel of the rest of the applications of the operating system used.

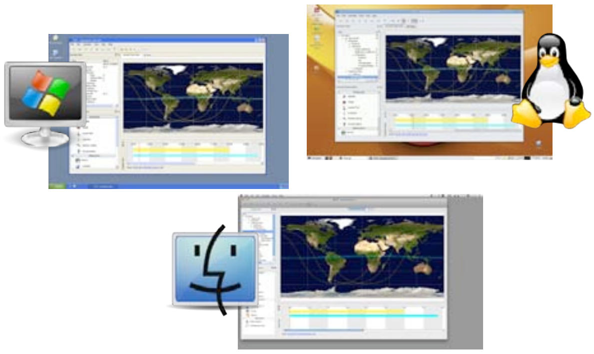

Figure 2. Running on Windows, Linux, and MAC OS X

STA has been tested to work on the following variants of the 3 operating system families supported: Windows XP, Windows VISTA, and Windows 7 (figure 5), Linux Ubuntu 8.10, Linux OpenSUSE 11.1, Linux Fedora 10, Linux OpenSUSE 11.2, Ubuntu 9.1, Linux Mint KDE 7 Gloria, (figure 3), MAC OS X 10.4, MAC OS X 10.5, MAC OS X 10.6 (figure 4).

All versions of STA require that the user has OpenGL installed. Nowadays, Linux, Windows and MAC OSX come with OpenGL drivers pre-installed.

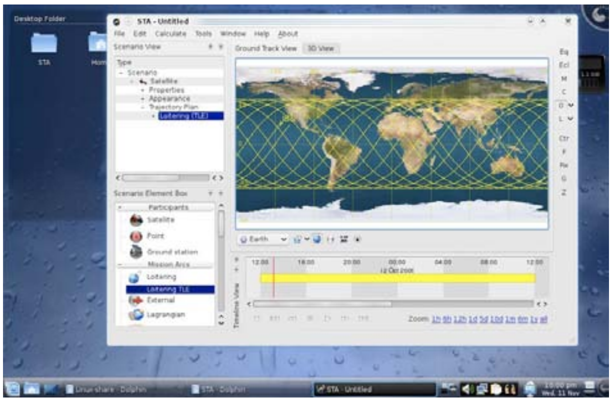

Figure 3. STA on Linus Mint KDE Gloria

Portions of STA can be controlled by scripts if the application is linked with libraries for the LUA language [R2]. Those come with the binary installations of STA in Windows, MAC, and Linux.

In general, to run STA, the computer should have a CPU (processor) that has a speed of at least 800 MHz. Typically, computers bought new within the last 4 years have adequate CPU speeds.

The 3D visuals of STA require an OpenGL compatible graphical card to run the 3D visualization graphics.

All code of STA is open source. STA uses also open source code provided by others.
STA uses Celestia code and Celestia data [R3]. In particular, STA uses Celestia's rendering engine for the 3D view window, some of its catalogs of stellar and planetary information, and some of its texture maps and 3D models of solar system bodies. STA uses the Eigen libraries. Eigen is a C++ template library for linear algebra that includes vectors, matrices, and related algorithms [R4].

Figure 4. STA on MAC OS X 10.6

For plotting, STA also uses Curveplot. Curveplot is a library for high precision plotting of piecewise cubic curves in OpenGL. STA uses the GLEW library to access advanced OpenGL features used in the 3D visualization. STA uses as well the QwPlot3D programming library that provides 3D-widgets for plotting. STA uses the optimizer NSGA-II from the Kanpur Genetic Algorithms Laboratory (KanGAL) of the Indian Institute of Technology Kanpur.

STA is compatible with the NASA SPICE system for spacecraft and planetary information. SPICE was developed by Caltech/JPL under contract to NASA [R5]. The star database (stars.dat) is derived from the ESA's HIPPARCOS star catalogue and it is the same that Celestia uses.

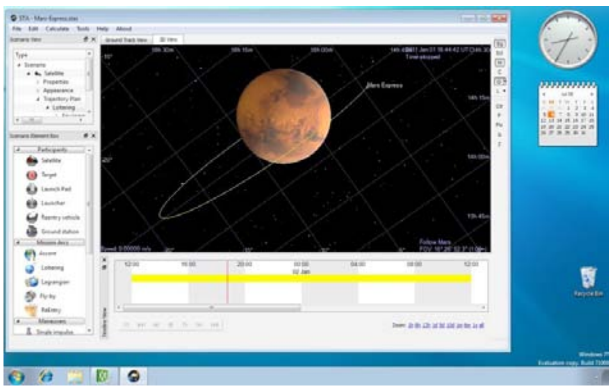

Figure 5. STA on Windows 7

On the memory and storage side, it is required to have a computer with a minimum of 128 Mb RAM and 500 Mb hard disk.

3. STA PARTNERS and TEAM

As research and education software applicable to Academia, a number of Universities and Institutions support this development partnering with ESA.

STA partners are organized around the STA Steering Board. Each participant member has a chair on the Board.

Currently, the STA Steering Board (figure 6) is composed of the European Space Agency, the Technical University of Delft (Netherlands), the University of Bremen (Germany), the University of Wuerzburg (Germany), the University of Kent (United Kingdom), the University of Coimbra (Portugal), the Complutense University (Spain), the University of Birmingham (United Kingdom), the Instituto Superior Técnico de Lisboa (Portugal), the University of Southampton (United Kingdom), the Technical University of Madrid (Spain), the Politechnico di Milano (Italy), the Institut Supérieur de L'Aéronautique et de l'Espace (France), the Technical School of Aeronautics (Spain), the Von Karman Institute (Belgium), the Technical University of Catalunya (Spain), and the Celestia team (USA).

Figure 6. STA Steering Board

The Steering Board has the duties to develop, promote, maintain, and expand the software suite. The Steering Board promotes and encourage the use, development, and maintenance of STA by means of publications, and a web presence. The STA Steering Board has the duty to propose the involvement of students in support of the development of STA: stages at ESA of Industry, Master theses, and PhD theses, etc.

The Steering Board meets every calendar year. The agenda of the Steering Board includes among others the following topics: status of the STA project, number of active developers, current members of the Board, etc.

4. USE OF STA and STA USERS

The modus operandi of the STA software development is to involve partners in the design, development, testing, and validation of its modules. STA uses as part of its workforce students developers from the partnering Institutions. Depending on their course background, students of aerospace engineering, mathematics, physics, computer sciences, or related are invited to contribute to the development teams from the partners universities.

In general, the STA project structure accommodates for Steering Board members, and student's development contributions via the following activities:

- **ESA contribution**: This way of involvement is a main way of contributions to STA. ESA staff lead, document, coach, develop, and manage code, students, relations, needs, and objectives on the STA project.

- **Master thesis projects**: This way of involvement ensures continuous and focused contribution from dedicated students while doing a Master thesis. During the time of the thesis work, the student is the developer and user of certain features or functionality of the STA software suite. All tasks are organized to allow a successful completion of the activities including verification and documentation in compliance with the STA development standards. In order to achieve this, the development of STA modules are bundled activities into thesis teams. The work conducted is supervised by the hosting partner university, and jointly together with ESA.

- **PhD thesis projects**: This way of involvement ensures a high innovative contribution from dedicated students while doing a Doctoral Thesis. As well as in the case of the Master student, during the time of the doctoral thesis work, the student is the developer and user of certain features or functionality of the STA software suite. In this case also, the work conducted is supervised by the hosting partner university, and jointly together with ESA and an Industrial Partner.

- **Internships**: Student developers can also contribute in the form of internship assignments when they are at ESTEC or from their parent Universities. This allows a more active interface with the personnel at ESTEC. During this time, the student benefits from shared supervision and support by the parent university and also ESA staff.

- **Space Mission Studies Team Members**: These are large space project studies in phase 0 or phase A that are complex to analyze and solve (e.g. mission to the Moon, mission to Titan, etc). The corresponding mission descriptions are released from ESA to the Steering Board for fulfillment. The analysis of these missions are done by students from several universities grouped in teams. These students divide the work typically along different mission arcs. The student teams improve and develop new STA modules as to be able to perform the requested analysis. Development activities that arise out of larger spacecraft mission studies (e.g. system engineering courses at universities or studies, initiated by space agencies) fulfill the same development obligations as single interns or thesis students. In this sense, these developers also require supervision by the hosting university.

- **Research Professionals**: These are developers and contributors who are on a more senior level, such as professors, research associates and experienced space engineers. Those form another category of STA developers. Due to their substantial background, professionalism and the longer extent of their commitment, they are capable of tackling advanced research responsibilities. In this sense, they are more likely than students to use STA as a test-bed for new algorithms. Research professionals also act as supervisors for the other three developer groups and commonly perform co-ordination responsibilities arising from steering board obligations.

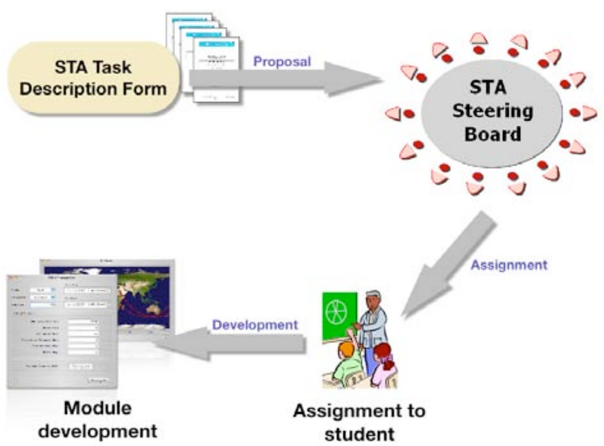

Figure 7. STA software development

5. YEARLY WORK on STA

The STA versioning system names are taken from the Earth geological EONS. “HADEAN” corresponds to version 1. “ARCHEAN” corresponds to version 2. The version 3 is named “CAMBRIAN”. Intermediate versions are released every quarter in between major releases.

The working cycle of the development of STA is synchronized with the time processes of the different STA partners. Every year a new version of STA is issued and the preparations for this event are planned along a yearly cycle that is depicted in figure 7.

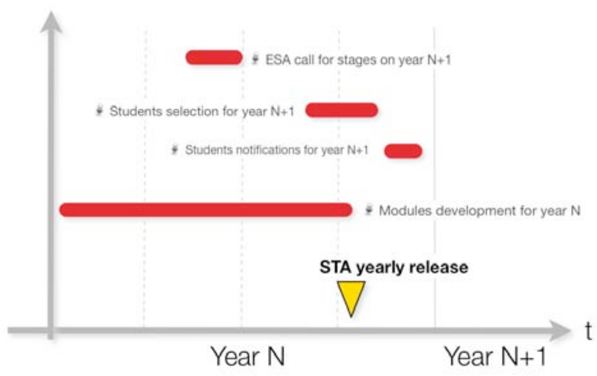

Figure 8. STA students team yearly cycle

In quarter 2 (QT2) around end of June, ESA calls the Steering Board for students proposals by releasing Task Description Forms TDF (equivalent to a user’s requirement specification). The release of the TDFs are issued together with a call for students nominations.

Students can be nominated either to be placed at ESTEC or to engage in STA tasks externally supervised by a University tutor, Institution tutor, or at ESA.

In QT3, students can apply for an ESA stage to perform at ESTEC the tasks described and agreed in a TDF. Also in QT3, external students can also apply for an agreed TDF to perform at their University the tasks described and agreed in the TDF.

In QT4, both ESTEC and external students are selected and nominations are announce as to start work from January next yearly cycle (figure 8).

Students assigned to work in STA are selected among the following types of Faculties: mathematics, aerospace, and informatics. Students are selected based on performance, dedication, motivation, and job description relevance.

6. USING STA: HOWTO

The functionality of the current STA version allows to create space scenarios, compute trajectories, create plots, 2D and 3D visualizations, produce a system level analysis of a space mission, and store data.

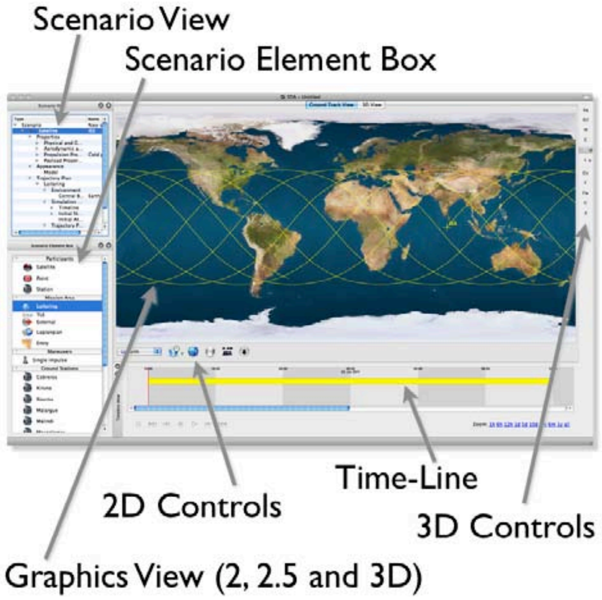

Figure 9. STA working areas

The current version includes the following mission arcs: re-entry trajectories, descent and landing trajectories, Lagrange point trajectories, loitering trajectories around planets and moons, and system engineering analysis.

Figure 9 shows all different working regions of a typical STA screen. The main screen contains 5 areas:

- The SCENARIO VIEW area. This area allows to add and remove all the elements that form part of a given scenario. The sum of all elements for the current scenario view.
- The SCENARIO ELEMENT area. This area contains all allowable elements of a given scenario, like spacecraft, ground stations,
- The RENDERING area. This area displays 2D and 2D trajectory visualizations of the elements that contain the scenario.
- The TIME-LINE area. This area displays and controls the time of live of the elements of the scenario. It allows to play back the movement of the spacecraft in the scenario.
- The CONTROLS area controls certain visualization features of the 2D and 3D rendering area.

In STA all working areas mentioned earlier can be docked and un-docked from the main rendering area (see figure 10). This feature allows the user to maximize the screen real state when concentrating on a particular area or type of work.

To undock an area from the main rendering area it is enough to click twice on the upper frame (banner) of the

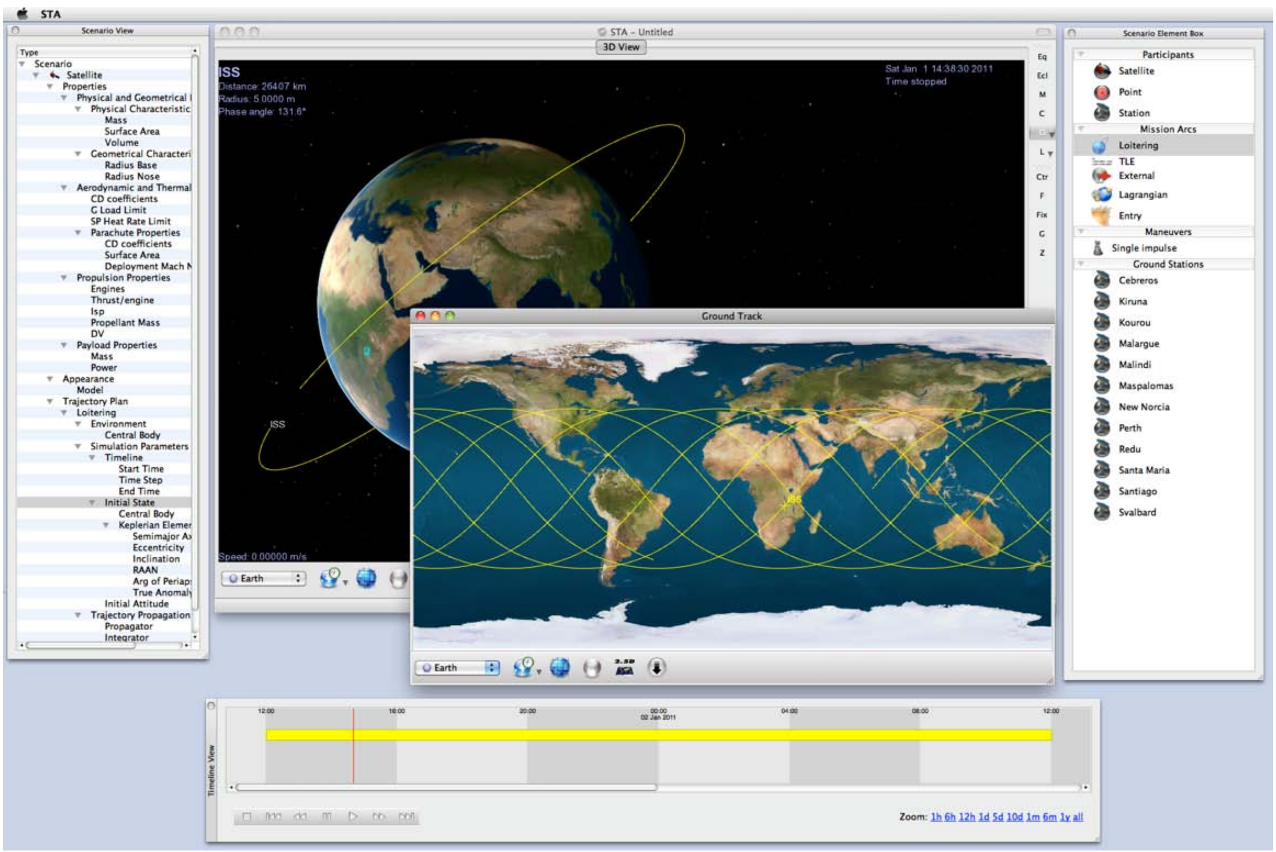

Figure 10. STA undocked view

area the user wishes to undock. It is also possible to undock an area by clicking on the undock icon. To dock back the undocked area to its original position, it is sufficient to click twice on the upper frame (banner) of the area the user wishes to dock back.

The rendering area undocks in a different manner: the user must explicitly click on the pull-down menu “Undock Ground Track View”. This produces the undock of the 2D view window from the 3D view window. To dock back the window, the user must click again on the pull down menu “Dock Ground Track View”.

The “Time-Line” area allows the user to control the movement of the space vehicles on a given scenario by playing-back a previously propagated session with space vehicles, ground stations, etc.

The time-line displays a set of bars in different colours. Each bar corresponds to a space vehicle (see figure 5.6).

7. SPACE SCENARIO

STA implements the concept of Space Scenario Paradigm (SSP). The SSP of STA implements the central idea of a scenario in which elements can interact to allow the user detailed analysis of these elements. STA defines the concept of Space Scenario (SS) as “a set of elements that play a role in the analysis of a trajectory path in a given Universe”.

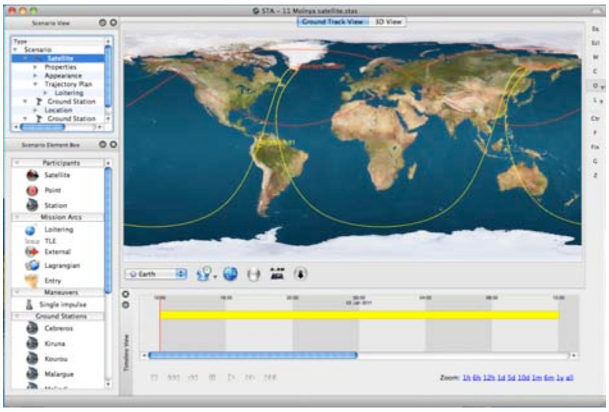

Figure 11. Scenario of a Molnya orbit

Following the previous definition, the components of a Space Scenario could be active participants (e.g. spacecraft, ground stations, sensors etc.) and also planets, aerodynamics database of a vehicle, co-ordinate systems, etc.

8. DOCUMENTING STA

One of the main interest on the development of modules in STA is the documentation.

Hence, a special emphasis is placed on teaching students and STA supporters on how to produce, and deliver documents that conform to the ECSS standards for the development of a complex software project. This is much appreciated in space industry and the students learn not only to write in technical English but also to conform to international standards widely enforced by ESA, European national Agencies, and its Industrial partners.

Starting from system level documentation, the STA Archean version introduced a totally new refurbished systems of document handling. The Task Description Forms written by the STA Steering Board constitute the User’s requirements document.

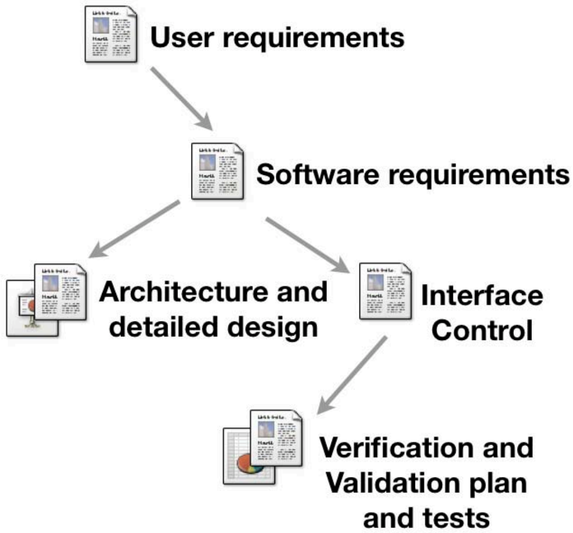

Figure 12. STA modules documentation tree

The person or persons responsible for the implementation of the TDF are in charge of writing the SDR (Software Requirement Document), the ADD/ DDD (Document of the architecture of the task), the ICD (Interface Control Document that will allow to integrate the software inside the rest of STA code), and the VVP (Verification and Validation plan of the code).

The Figure 12 depicts this tree structure. On addition, the implementer(s) are required to deliver as well the SUM (Software User's Manual). The SUM will be transformed later on onto a Wiki page and uploaded to the STA Wiki.

STA provides templates for all documents following ECSS tailored standards. All STA documentation is written using OpenOffice.

9. OPEN SOURCE

The entire STA code is released open source. Open source has been considered the best possible vehicle to motivate students to work in the areas of astrodynamics, mathematics, and informatics.

The model of open source allows as well to increase the quality of the software and also its awareness. It helps also in its maintenance and operational life.

According to the FLOSSMetrics (Free/Libre Open Source Software Metrics), the estimated cost to develop this project is about 6 Million Euros. Up to date, STA has 238.709 lines of code. This includes code from other open source projects under which STA relies. According to FLOSSMetrics estimations [R9], to develop a similar application the team would need 62,79 person-years, 2,58 years with an average of 4,3 number of developers.

The STA project is free software; anybody can redistribute it and/or modify it under the terms of the European Union Public License EUPL v.1.1 [R6] as published by the European Commission. STA is distributed in the hope that it will be useful, but without any warranty.

10. STA FUNDING

Since STA is an open source project, there is no direct funding associated with it. Students participate voluntarily in the hope of obtaining a grade and hence are funded by Universities. Professors are also paid by Universities and Institutions.

While the STA modules are developed by the partnering Universities, ESA chairs the STA Steering Board and acts as secretary. ESA also coordinates the integration of the modules produced by the students in the final framework. ESA provides also a static web site as Portal of the project.

The STA project is not funded by any ESA study (GSP, TR, GSTP, etc). This is much different from some ESA open source initiatives like GEANT-4 (funded on GSTP), SPENVIS (GSP, TRP) etc. However, ESA provides internships at ESTEC to develop some of the modules.

11. CRITICISM

While the involvement of students as software development work force is a beneficial concept for the STA project, its has some drawbacks too.

Students are engaged on topics of direct interest for Universities. Some times, the requested topics are not fully inline with the development plans of STA. In these occasions negotiation ends in compromises for the development of the software.

Another drawback is the availability of students in particular technical fields. Some tasks are awaiting the proper student profile to be fulfilled while in some occasions too many students are available for a similar more attractive topic.

Also, the development of STA is slow. As an open source project, the workforce is not constant and produces time gaps where the code updates or maintenance is not happening.

12. WHERE TO FIND MORE

The web site of STA is as follows:

http://sta.estec.esa.int

This web site contains all information required to inform the users and developers, to be able to download the code, and to gather information about the development status.

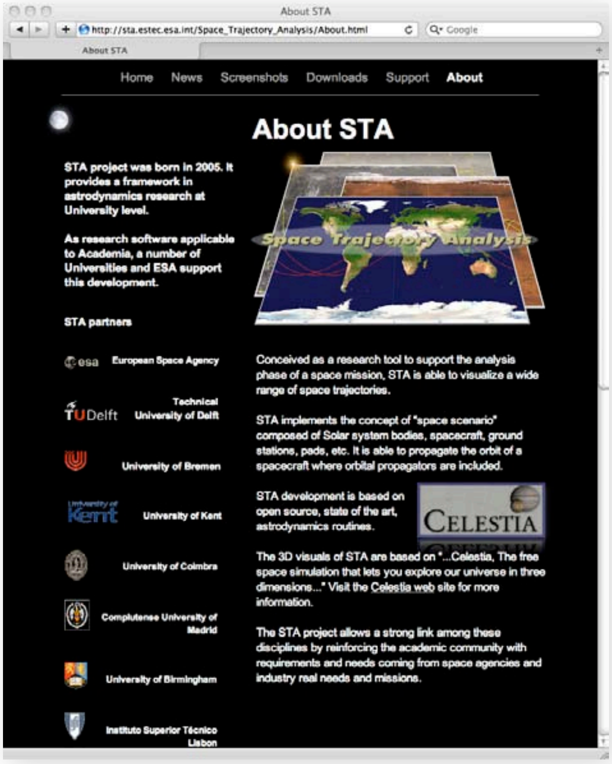

Figure 13. STA web site: about view

To be able to co-ordinate a wide range of people involved in the STA project with many lines of code, several countries, places, etc, STA implements a good system grouping several development resources. All these resources are grouped into the term of “STA data”.

STA data is formed by the following elements: source code, documents, promoting material in the form of web site pages, wall papers, posters, newsletter, etc. Also material in the form of forums, news groups, e-mail communications, etc. is considered information related to the STA project.

The data pertaining to the STA project is stored in several centralized repositories.

The STA repositories of source code are stored using the services of OSOR. OSOR has the largest repository of open source code and applications available in Europe on the internet. OSOR site provides a number of web-based tools to help the development of open source projects [R7].

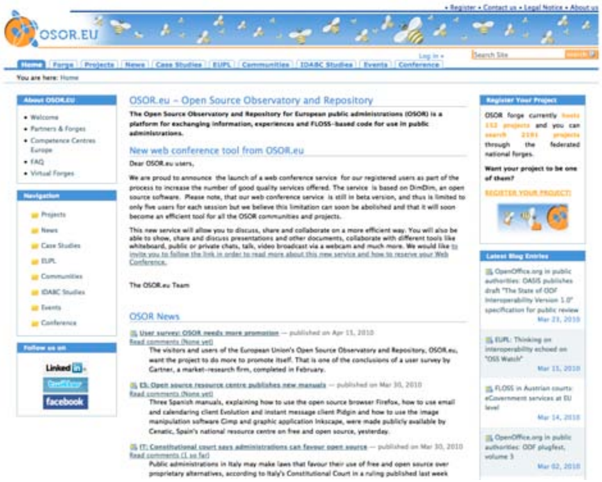

Figure 14. STA hosted on OSOR

13. ACKNOWLEDGMENT

The authors of this article would like to thank the invaluable work of Margaret Hendricks (ESA) and Audrey Paterson (ESA) on support to the STA stagiaires at ESA. Special thanks to the Head of the Control Division at ESA, Alain Benoit for his constant support to STA.

14. REFERENCES

[R1] ECSS software requirement standard ECSS-E40. Web site as per the following URL: http://www.ecss.nl/

[R2] The LUA programming environment can be found at this URL: http://www.lua.org/

[R3] Celestia User's Guide, Rev 1.4.1 and 1.4.0, February 2006. Celestia web site as follows: http://www.shatters.net/celestia/

[R4] The Eigen libraries. Web site as per the following URL: http://eigen.tuxfamily.org/index.php?title=Main_Page

[R5] The NASA SPICE ToolKit. Web site as per the following URL: http://naif.jpl.nasa.gov/naif/toolkit.html

[R6] The European Commission Public License version 1.1. URL as follows: http://ec.europa.eu/idabc/eupl

[R7] The OSOR European Commission open source software repository. URL as follows: http://www.osor.eu/

[R8] The Qt development environment. URL as follows: http://qt.nokia.com/products

[R9] FLOSSMetrics stands for Free/Libre Open Source Software Metrics. URL as follows: http://flossmetrics.org/