ESTABLISHING A COMMUNITY APPROACH TO SST CORE SOFTWARE THROUGH A USER FORUM

Johannes Klug(1), Hélène Ma(2), Tim Flohrer(2), Elmar Brendel(1), Efrén Villanueva(3), Jesús Tirado(3), Anabel Mediavilla Garay(4), Christian Unfried(5), Matti Nylund(6), the SST-UF

(1)OPS-GDA, ESA/ESOC, Robert-Bosch-Strasse 5, DE-64293 Darmstadt, Germany,
Email: johannes.klug@esa.int, elmar.brendel@esa.int
(2)ESA Space Debris Office, ESA/ESOC, Robert-Bosch-Strasse 5, DE-64293 Darmstadt, Germany,
Email: helene.ma@esa.int, tim.flohrer@esa.int
(3)GMV, Calle Isaac Newton 11, Tres Cantos, 28670, Spain, Email: evillanueva@gmv.com, jtirado@gmv.com
(4)Deimos Space, Ronda de Poniente 19, Tres Cantos, Madrid, 28760, Spain,
Email: anabel.mediavilla@deimos-space.com
(5)Space Analyses GmbH, Marxergasse 24/2, A-1030 Wien, Austria, Email: christian.unfried@spaceanalyses.at
(6)Swedish Defence Research Agency, SE- 164 90 Stockholm, Sweden, Email: matti.nylund@foi.se

ABSTRACT

Since the start of ESA's Space Situational Awareness (SSA) Programme in 2009 the number of European activities has been growing significantly. Beside today's dominant national and multi-national efforts to build-up and maintain object catalogues and to derive services, commercial endeavours are also starting. In support of the European partners ESA introduced the concept of a community approach towards a Space Surveillance and Tracking (SST) Core Software in 2017 [1][2] as a key objective of ESA's SSA Programme, which from 2020 has evolved into ESA's Space Safety Programme.

This community approach established the SST User Forum as the community's technical advisory body for the SST Core Software, which is composed by several ESA Member States participating with their national representatives to support the community needs.

Several activities are being conducted in the development of the SST Core Software. The initial software baseline (v0a) was created from previous developments under the SSA Programme, and has been approved for licensing to the SST-UF members under ESA Community License in 2020. The upcoming version of the software will be automatically built, tested and deployed using Gitlab Continuous Integration and Continuous Development pipelines and Docker, under a Collaboration and Software Development Framework.

We here provide an overview of the community approach and the current status of the SST Core Software, outline the collaboration framework supporting the community through the SST User Forum in the software development, maintenance and evolution, and report on the related work-plan activities in ESA programmes.

Keywords: SST Core Software; Software Community Approach; SST User Forum; ESA's Space Safety Programme.

1 INTRODUCTION

Space Surveillance and Tracking (SST) addresses detecting, cataloguing, and predicting objects orbiting the Earth, and the derived applications, such as conjunction prediction, re-entry prediction, and fragmentation analysis. SST technology development, test and validation represent one of the main segments in ESA's Space Situational Awareness (SSA) Programme. Since 2020, the SSA Programme is expanded by the Space Safety Programme (S2P) that conducts activities in the areas of Space Weather, Planetary Defence, and Space Debris and Clean Space. The establishment of a community approach to the SST Core Software is pursued within S2P, directed towards the goal of a European autonomy in accessing and using space in a safe and secure environment.

ESA's SSA Programme Participating States have and operate their own SST assets within a European SST landscape that is evolving rapidly and significantly. Commercial activities in the SST domain have begun to increase, e.g. for sensor data acquisition and processing and derived SST services. Within the European partners support, the SSA Programme developed the concept of a community approach towards the SST Core Software in 2017 to provide interoperability, compatibility and harmonisation in system outputs and services provision of the various developments in Europe, while allowing closed-source national tailoring according to specific needs.

The paper will provide a description of the SST Core Software and outline the concept of the community approach through a User Forum with the Collaboration and Software Development Framework, and on the related work-plan activities.

2 SST CORE SOFTWARE

The SST Core Software is a set of tools designed as a whole system covering the community needs in SST activities: it provides a set of services and applications from the interface handling of SST sensor data and third-party orbital data, including the sensor planning and tasking, the detection, identification, cataloguing, and monitoring of the artificial objects orbiting the Earth, the prediction of conjunction and re-entry events, and the analysis of fragmentation.

This toolset is defined as a composite of subsystems implemented with state-of-the-art technology in order to provide a complete and comprehensive aggregation of functionalities to process different types of sensor data (radar, passive optical and laser observations) into data and services required by the SST users.

The initial baseline of the software was created in previous developments in the frame of the SSA Programme, i.e. elements for the surveillance and tracking data processing - including functions for scheduling, orbit determination and correlation - as well as service output generation. The current baseline v0a is the final delivery of the activity P2-SST-V-VI-VIII "SST Data Centre Integration and Enhancement" [3] whose primary objective was to integrate the software subsystems into a fully functional end-to-end system. A set of functional gaps was implemented among the potential improvements that were identified from previous development and integration campaigns, and will be applied within the following activities. The next version of the Core Software is currently under development within the activity P3-SST-IX "SST: Core Software Requirements and Framework Finalisation", and will be automatically built, tested and deployed using Gitlab Continuous Integration and Continuous Development (CI/CD) pipelines and Docker. A Software Development Framework has been created also within P3-SST-IX, involving a complete set of tools to handle requirements, issues, release management, document storage, software version control, source code analysis and other CI/CD tasks.

This software is not classified. Data processed through the SST Core Software will not be incorporated in the releases but ESA plans to make unclassified reference data available with the SST Core Software for test and validation purposes.

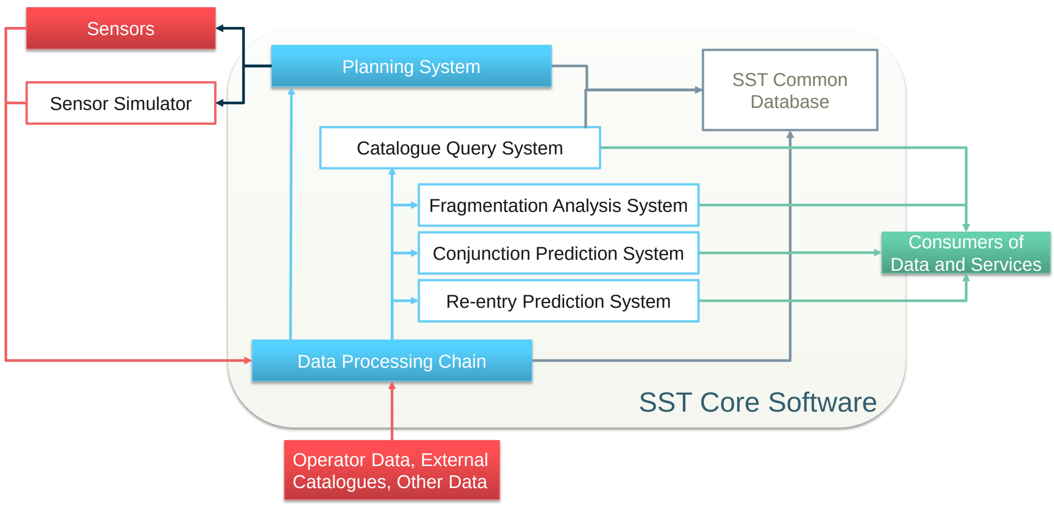

Figure 1: High level architecture of the Core Software

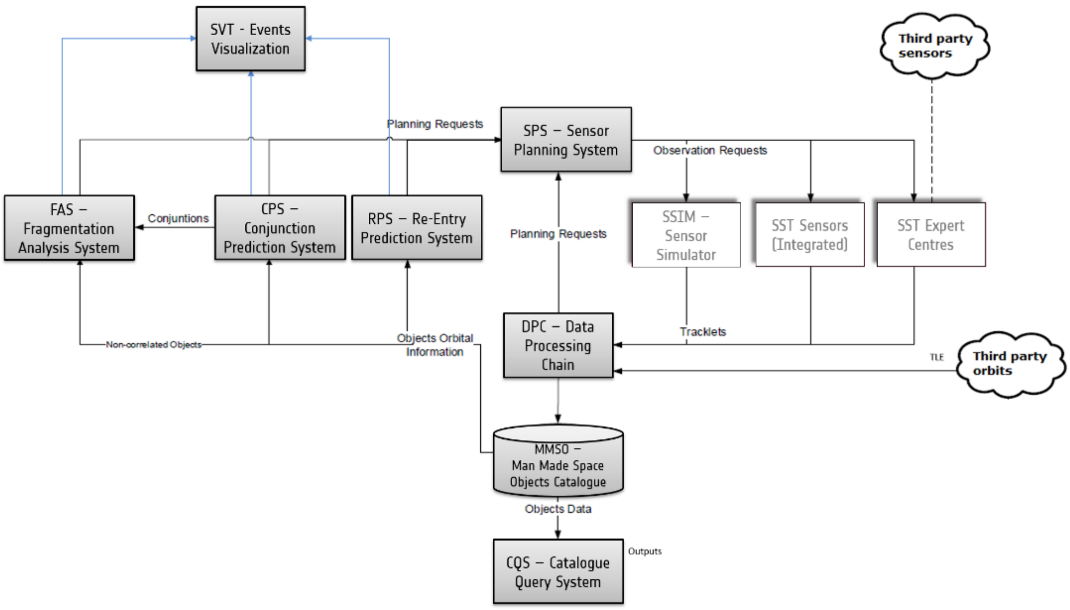

Figure 2: Architecture of the SST Core Software at system level

2.1 System components and functionalities

The architecture of the SST Core Software is represented in Fig. 1 and Fig. 2. The main functionalities of the system are achieved through the following components:

- Data Processing Chain (DPC) contains the main modules to perform orbit determination, orbit propagation, data correlation, and orbit accuracy assessment using the observation data received from the sensors. It generates and maintains the catalogue of detected objects, the Man-Made Space Objects Catalogue (MMSO), by requesting survey and tracking observations.
- Sensor Planning System (SPS) receives planning requests from the services, the DPC or system operators and optimises the observation schedule for a given sensor network and cataloguing needs.
- Catalogue Query System (CQS) provides access to the MMSO, generates reports with information retrieved from the catalogue and notifies users about the presence of updated object information.
- Conjunction Prediction System (CPS) detects conjunction events between the catalogued objects, generates warnings and requests new observations of objects involved in high-risk conjunctions according to the user- or system-defined thresholds to the collision probability and the miss distance.
- Re-entry Prediction System (RPS) analyses the catalogued objects' lifetime, predicts re-entries and requests new observations of objects involved in risk object re-entries.
- Fragmentation Analysis System (FAS) detects and characterises in-orbit fragmentations and the source of the objects resulting from those events. It can use predicted conjunctions from the CPS component to assess if a fragmentation emerged from a collision event.
- SST Visualisation Toolbox (SVT) is a graphical tool that is specifically designed as an user interface to visualise in 2D and 3D the SST events predicted by the different services CPS, RPS and FAS as well as the objects and their obits stored in the MMSO catalogue.

The MMSO, and the databases for the sensors, SPS, DPC and the respective SST Services are contained into a central and single Postgres database called the SST Common Database which was designed to homogenise its usage by the different subsystems. The pillar of data production and processing is the Planning – Observation – Data Processing loop. Different assets contribute to data refinement and allow the generation of data products that are targeted for the prediction of conjunction and re-entry events, and for the analysis of fragmentation. These services are complemented by a set of front-end subsystems, in particular the CQS that queries the catalogued objects information to the SST Common Database, and the SVT giving visual aid for the events and object orbit.

In the front end, the functionalities of the services CPS, RPS, FAS and CQS can be accessed by a Web Portal (see Fig. 3) or an SST services Human-Machine Interface (HMI). The SST events can be visualised by the SVT from a Web browser (Fig. 4). In the back end, the functionalities of the DPC and SPS are accessible through an HMI.

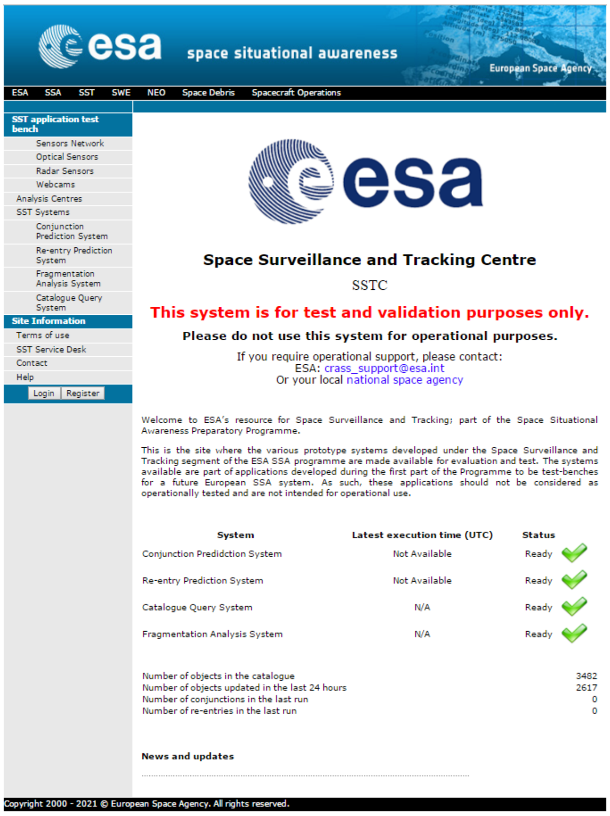

Figure 3: Main page of the SST Web Portal

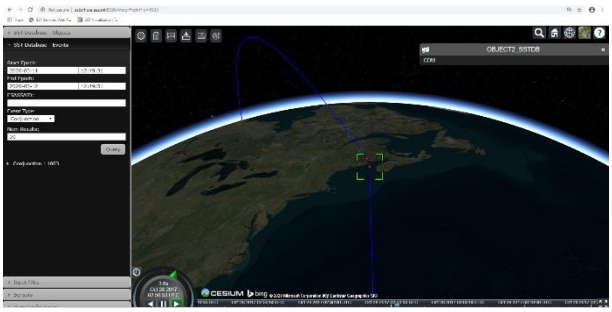

Figure 4: A screenshot of SVT

Other external entities can be integrated into the SST Core Software, such as a Sensor Simulator, external individual sensors directly, or an Expert Centre. These different modules are explained in the following paragraphs.

The Sensor Simulator can allow to simulate sensor observations and test the performance and capabilities of the Core Software with data at larger volumes.

Individual sensors are also expected to work autonomously and to develop capabilities to interact with the Data Processing loop.

The Expert Centre system acts as a proxy between external heterogeneous and distributed sensors (passive optical sensors and satellite ranging stations) and a data processing backend, i.e. DPC system of the SST Core Software. Therefore, it diminishes the overhead in case of coordinated observation campaigns by centralising observation requests, by assessing sensors status and Key Performance Indicators (availability, latency, epoch bias, response time etc.), and by monitoring Service Level Agreement compliance. Moreover, it can provide significant added-value expert support and calibration services to external sensors, data quality assessment and data exchange standards compliance verification [4][5].

2.2 System modularity

The complete system is designed in order to facilitate future additions and tuning of the software to specific user needs. The present architecture in Fig. 2 also allows appending state-of-the-art methodologies and tailoring the system to individual needs.

All the subsystems apply the existing standard data formats used in SST activities like the “Consultative Committee for Space Data Systems” (CCSDS) to ensure consistency in the data exchange with external modules, in the common development of the subsystems and interfaces, hence in the modularisation of the software.

3 THE COMMUNITY APPROACH THROUGH A USER FORUM

The capabilities of the SST Core Software are consolidated by the concept of the community approach. The SST Core Software community will be composed of stakeholders and users of the software.

The SST User Forum (SST-UF) represents the community’s technical advisory body of the SST Core Software. As of now, it is composed by the following Member States that are participating with their national representatives to support the community together with ESA: Austria, Germany, Finland, Italy, The Netherlands, Norway, Poland, Portugal, Spain, Sweden, Switzerland, and the United Kingdom. All Member States that subscribe to ESA’s Space Safety Programme can propose one technical representative for their participation in the SST-UF.

The SST-UF (including ESA) has been meeting twice a year since November 2017. During the meetings the status of the Programme activities related to the developments leading to the Core Software have been a standing agenda item, as well as briefings on developments by Participating states in the domain. The ESA Executive also conducted demonstrations to SST-UF members on the software status and capabilities, and workshops addressing specific topics areas.

The primary purpose of the SST-UF is to:

a) facilitate collaboration in SST by forming a community that provides and maintains the main building blocks and reference implementations for SST systems (the “SST Core Software”);

b) determine the boundaries of the SST Core Software and the overall direction of the community's work;

c) support the management of the requirements baseline, change control, release management, and the management of the industrial contracts;

d) promote and support the use of the SST Core Software;

e) establish procedures for testing and validating in a coherent and comprehensive SST environment;

f) develop and maintain procedures for qualifying SST sensors; and

g) address the required standardisation efforts for SST data exchange formats.

The version v0a of the Core Software has been checked for licensing, i.e. absence of background intellectual property in the release. In 2020, the ESA Software Licensing Board has approved that the SST Core Software can be released under ESA Community license strong copyleft, v 2.4 to the Member States participating in the SST-UF. Licenses can be granted to the SST-UF Point of contact, and sublicensing is possible.

A System Requirements Review (SRR) for the Core Software was held to collect user needs and stakeholder interest through a series of interviews. The SST-UF members have been invited to comment on the SRR and findings from the discussions have led to the recording of current functional and performance gaps that are being prioritised and addressed in ongoing and upcoming developments.

The SST-UF is currently asked to comment on the design of the collaboration framework, which sets the future processes for deciding on feature implementation and handling of change requests, problem reports, and non-conformances with the agreed baseline. As long as the Core Software source code is being hosted by ESA, it will be possible to use ESA's own Gitlab infrastructure. Collaboration on this platform is possible for users with approved user accounts via internet.

For this reason, the SST-UF plays a key role in providing guidance on all aspects of the software evolution, covering the identification and consolidation of requirements and their implementation. It also supports the governance and licensing approach and contributes to the standardisation efforts for SST data exchange formats. For instance, a recent workshop with a Splinter group of the SST-UF has been initiated to enhance the definition of the detailed governance model.

Consequently, the user community in the SST-UF is in the position to drive and control future releases of the software. It is also responsible for defining and managing the technical and organisational goals of the collaboration mechanism, including the workflows for the SST Core Software Development Framework.

4 COLLABORATION AND SOFTWARE DEVELOPMENT FRAMEWORK

4.1 Continuous Integration & Continuous Development (CI/CD)

To support the community development approach, a Software Development Framework has been created following the CI/CD philosophy. It includes a complete set of tools to handle requirements, issues, release control, document management, software version control, analysis of source code and other CI/CD tasks.

CI/CD is a cultural shift for developer teams, and a collaboration concept reflective of key aspects of software development and delivery processes. This philosophy is employed for the SST Core Software development to facilitate broad user requirements and to be used by a large extent of user communities (e.g. academic institutions, SMEs, industry and agencies). Each valuable contribution and requirement set is taken in through CI/CD.

4.2 Software development framework

The Software Development Framework that was created within P3-SST-IX is based on Gitlab. Gitlab itself is a fully-featured development platform based on the git version control system. ESA hosts its own Gitlab instance (gitlab.esa.int), which is used successfully by hundreds of projects within the organisation. For the Space Safety domain in general, and Space Debris in particular, this platform is also heavily used and the experience has been very positive overall. Adopting the same platform also for the SST Community Approach was a natural choice. Out of the box, it supports CI/CD, and lends itself well to distributed developments, thanks to the underlying git version control system. The CI/CD workflow is shown in Fig. 5.

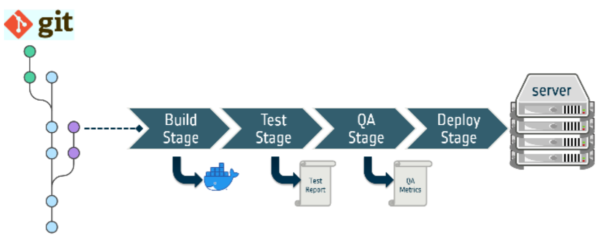

Figure 5: CI/CD workflow

4.3 Governance model

Defining the software governance mechanism for the SST Core Software is essential at this stage of the development but it is also challenging as the system is an integration of several subsystems implemented from previous developments, and the community concept seeks to establish and sustain a reliable Collaboration and Software Development Framework.

The national stakeholders, hence the SST-UF members, are the most suitable actors to build the governance model, as they can drive its definition from the needs of their respective national communities.

The governance model of the SST Core Software under the Collaboration and Software Development Framework is currently under elaboration with the SST-UF.

The aim is to arrive at an agreed workflow for collaborative development. All UF members should benefit from a commonly developed baseline, while at the same time having the option to pursue national developments that add missing features for their national use cases. While it is encouraged to feedback changes as much as possible to the community, it is understood that this might not be desired or even possible in some cases. The governance model shall reflect this accordingly.

5 OUTLOOK

At the time of writing, the details of the SST-UF Governance Model are under discussion within the User Forum. It is therefore still too early to discuss the identified stakeholders, their roles and the processes linking it all. This will be summarised in a dedicated report once the governance model has been agreed and practiced for some time.

In continuation of the former SSA Programme, the established work plan of the new Space Safety Programme has already been initiated.

The P3-SST-XXVI activity "Core Software Development Baselines and Architecture Modernisation under Community Approach" was recently started. It aims at enhancing the SST Core Software baselines addressing the SST-UF needs and derived requirements, including the improvement of the architecture, the modularity characteristics, the performance and the standardisation.

The P3-SST-XXVII activity "Customised SST software elements in the SST Core Software and Expert Centre" is being conducted in parallel with P3-SST-IX and P3-SST-XXVI. It will demonstrate how national tailoring can be used in the SST Core SW community approach and in the development of national Expert Centre systems. Moreover, a large set of testing and verification data from optical, radar and satellite laser ranging sensors is being collected over a series of observation campaigns. Together with simulated data these resources are intended to support test and validation of the SST Core Software.

A subsequent activity to support the Core Software community by operating the Software Development Framework is foreseen under Space Safety Programme, i.e. via activity "S1-SC-08 Supporting the community for the Core Software development, maintenance, and standardisation". Within this activity, the community will be supported during 2021-2023 in using and testing the baselines, and in developing own contributions flowing back. It will also introduce a product management for the releases.

6 ACKNOWLEDGMENTS

The authors would like to thank the SST User Forum members for their contribution to this paper, for their active contributions to the User Forum in general, and the collaboration and governance model in particular.

7 REFERENCES

1. Space Surveillance and Tracking in ESA's SSA Programme, T. Flohrer and H. Krag, In: *Proceedings of the Seventh European Conference on Space Debris*, 18-21 April 2017, ESA/ESOC, Darmstadt, Germany.

2. SST-CC: A community Approach for a European Space Surveillance and Tracking Core Software Development, Christian Unfried et al., *6th European Ground System Architecture Workshop*, 20-21 June 2017, ESA/ESOC, Darmstadt, Germany.

3. ESA's SST Data Centre Integration, J. P. Arregui, J. Tirado, E. Parrilla, J. Castro, D. Escobar, A. Águeda, N. Sánchez-Ortiz, H. Demir, H. Krag, In: *Proceedings of the Seventh European Conference on Space Debris*, 18–21 April 2017, Darmstadt, Germany.

4. Expert Centres: A Key Component in ESA's Topology For Space Surveillance, B. Jilete, T. Flohrer, T. Schildknecht, C. Paccolat, M. Steindorfer, In: *Proceedings of the First NEO and Debris Detection Conference*, 22-24 January 2019, Darmstadt, Germany.

5. A Pan-European Expert Centre service and coordination facility in support of space surveillance, C. Paccolat, T. Schildknecht, M. Steindorfer, B. Jilete, T. Flohrer, In: *Proceedings of the Eighth European Conference on Space Debris*, 20-23 April 2021, Darmstadt, Germany.