REUSE OF OPERATIONAL FLIGHT DYNAMICS SOFTWARE FOR SPACE SURVEILLANCE

European Space Surveillance Conference
7-9 June 2011

Alberto Águeda(1), Fran Martínez(1), Jaime Fernández(1), Diego Escobar(1)

(1)GMV Aerospace and Defence, S.A.U.
C/ Isaac Newton, 11. PTM - Tres Cantos
28760 Madrid. Spain

Emails: aagueda@gmv.com, fmartinez@gmv.com, jfernandez@gmv.com, descobar@gmv.com

INTRODUCTION

The deployment of the Space Surveillance and Tracking (SST) precursor service is demanding a lot of infrastructure software for a wide variety of applications. Many of these applications are closely related to flight dynamics and are optimal in the reuse scenario intended by the precursor services of SST. GMV has been traditionally contributing to the development of flight dynamics related software for ESA and also a consumer of those developments to provide ad-hoc solutions in other related areas. From this prospective, this paper provides a view on the potential reuse of ESA and other parties’ software packages for different activities associated to the development of the SST precursor services and potential extension to the final SST system.

An initial collection of software packages has been identifies that fulfils some of the needs of the initial developments of the space surveillance services. This paper addresses among others:

- NAPEOS, the Navigation Package for Earth Orbiting Satellites developed by ESA that provides a comprehensive suite of tools for flight dynamics operations and comprehensive data analysis. NAPEOS is a good candidate for reuse given its modular architecture that even permits the development of additional functionality on its structured core library.
- CRASS (Collision Risk Assessment) and ODIN (Orbit Determination with Improved Normal Equations) are two packages providing conjunction analysis and ad-hoc orbit determination for SST. Their software approach is to some extent obsolete (from the software point of view) but the algorithms and core functions can be easily reengineered to be more efficient and maintainable. Both elements are operational at ESA and represent an excellent reference for the validation of new implementations.
- closeap is GMV’s evolution of CRASS to object orientation with an optimised implementation that allows parallelisation. It inherits the algorithms from CRASS being therefore compatible with ESA’s approach for collision risk assessment.
- focussuite is GMV’s solution for the development and integration of Flight Dynamics applications.

Note that all packages defined in these bullets are in one way or another related to ESA developments. One critical issue in the development of the SSA precursor services is the software availability and the potential dependency of that availability with software licenses and third party Intellectual Property Rights (IPR).

THE SPACE SURVEILLANCE AND TRACKING PROBLEM

First of all it is to be noted that in the following this paper refers to the SST as a precursor service within the SSA program. The final approach for the SSA design and deployment is not defined yet and may entail a completely different approach that does not imply reuse. For the precursor services the principle of reuse is a must and therefore what is described in this paper applies in its entirety.

The approach for deployment of the SSA services has been described in a number of occasions and can be summarised in Fig 1., which identifies the need to decompose the SSA services at high level into processes and low level services that can be combined in a number of ways. This paper identifies some of the low and intermediate building blocks to be reused from existing software that provide solutions for basic and application services that well established and operationally proven.
The solution intended for the SSA precursor services is that the different solutions for the elements in the different levels are integrated in a service oriented architecture (SOA) that implements a sort of ‘plug-and-play’ framework where service components can be deployed, enhanced and replaced without affecting the provided service to the users. This is the Common SSA Integration Framework (COSIF) where all services have to be integrated. It provides an integrated environment for developing, upgrading, testing, operating and maintaining the existing and new SSA software systems. At this point the first potential difficulty appears as the reuse principle implies that newly developed software coherent with SOA technology must integrate legacy software that in general is not prepared for the SOA type of integration.

SOFTWARE REUSE APPROACH

The objective of the analysis is to identify the software packages, libraries and tools that can be subject of reuse as they provide solutions for the functionality required for SST. Reuse is intended to benefit from the robustness and validation level of existing solution with having to perform again a whole software engineering cycle that implies risk and effort. Software maturity cannot even be gained after an exhaustive software engineering cycle.

Three main approaches for the reuse of software are identified, which are complementary and not mutually exclusive:

- Direct reuse, for cases where the package readily provides the required functionality with very minor adaptations (e.g. interfaces). This is the most desirable case but in general specific requirements imposed from the service levels will cause that this option is difficult to exercise in its entirety.

- Adaptation and enhancement when the package provides a core functionality but requires additional the implementation of functions to complete the expected functionality. This scenario also considers the case when the software needs to be reengineered to adhere to adequate software and algorithm standards.

- Validation when the package is not adequate for reengineering (e.g. too expensive or there is another better suited package) but the implementation provides reliable results that can be used as reference for other implementations.

There is a case that has not been considered in the previous bullets that corresponds to a complete refactoring of an existing package. From the point of view of this paper that would be considered as a completely new development, no reuse therefore. This situation can be mapped into the third case where the legacy software is used for the validation of the newly developed one.

REUSABLE SOFTWARE PACKAGES

As identified in the introduction, there are a number of software packages that provide extensive and well proven flight dynamics functionality that can be reused to implement SST applications either as a complete application or providing a robust infrastructure.

- **NAPEOS** [1][2] (Navigation Package for Earth Orbiting Satellites) is a flight proven and well-known operational ESA software used extensively along Europe for Flight Dynamics and high-performance Orbit Determination related operations (including the most demanding and accurate scenarios in all phases of an Earth orbiting mission).

NAPEOS implements state-of-the-art and complete force models for orbit propagation (POD capable, including propagation of covariance matrices), comprehensive measurements reconstruction patterns (for most tracking techniques including optical angular measurements –RA/DE– and radar measurements –range, range-rate and...
angles, AZ/EL-) and detailed measurements correction models (POD capable). High precision generic orbit determination algorithms are included in the system for both batch (weighted Bayesian least-squares with and without a-priori information) and sequential processing (SRIF). It fully supports the IERS-96 and the latest IERS-2003 standards. It additionally includes multiple tools for analysis such as reference frames rotations, transformations between reference time systems, orbit formats conversion (including CCSDS OPM and OEM orbital formats), orbits comparison, etc.

Due to its modularity and expandability, NAPEOS has been extensively reused for a diverse variety of other space applications, including cataloguing for space surveillance: SSASIM [3].

- **ODIN** [4] (Orbit Determination via Improved Normal Equations) is an operational ESA software operated by the Space Debris Office for the execution of orbit determination based on SSA tracking data from optical telescopes and radar. ODIN implements tracking pre-processing routines for several SSA systems in Europe (OGS, TIRA, ARMOR/MONGE), tracking simulation for optical and radar data, orbit propagation with detailed models, initial orbit determination for both angles-only and angles+range (ad-hoc implementation of the Gauss algorithm in both cases) and batch least-squares orbit determination (including standard Bayesian least-squares, rank reduction and Levenberg-Marquard algorithms). The interoperability between NAPEOS/SSASIM and ODIN is ensured by the existence of a number of tools in ODIN for the conversion of internal measurements and orbit formats.

- **CRASS** [5] (Collision Risk Assessment Tool) is an operational ESA software operated by the Space Debris Office for the detection of close encounters between debris objects and ESA payloads. CRASS is a generic purpose collision risk assessment tool (developed by GMV for ESOC) aimed at detecting conjunction forecasts and computing collision risk estimates and events for ESA satellites with respect to the full USSTRATCOM catalogue population (stored in TLE format) based on available covariance information. These events are computed by means of a **smart sieve** [7][8] technique consisting of a series of fast and robust filter algorithms.

- **closeap** [6] is GMV’s evolution of CRASS to object orientation with an optimised implementation that allows parallelisation. It inherits the algorithms from CRASS being therefore compatible with ESA’s approach for collision risk assessment. The main improvements in closeap with respect to CRASS are the integration in the NAPEOS structure, the optimisation of the algorithms for performance and object orientation with the target of parallelisation.

- **focussuite** [9] is GMV’s solution for the development and integration of Flight Dynamics applications. It is a generic client/server infrastructure that allows the integration of applications and that is also suitable for the integration with an Enterprise Service Bus (ESB) in a system implemented according to the Service Oriented Architecture (SOA) paradigm. focussuite provides two well defined and isolated elements, the HMI that may or may not be subject of any reuse depending on specific requirements and functional needs and the client/server infrastructure that provides data and process control and is subject of reuse for the integration of existing applications in the ESB of the COSIF (see next).

**INTEGRATION IN THE COSIF**

One of the principles of SOA is the encapsulation of functionality behind services that are exposed to the users. The COSIF implements this principle which is most relevant in the case in which legacy software is used to provide internally the exported functionality. One particularity of this principle is that the exposed service should be free of hardware or software couplings. However, the reuse of legacy software may imply tight couplings with hardware and system software that prevent the direct integration of the applications in the COSIF. This difficulty is circumvented by physically segregating the COSIF from the application elements. The COSIF is then deployed separately from the applications that are deployed in the application nodes that have specific characteristics. This has the following consequences:

- It is necessary to define the interfaces between the COSIF and the application nodes. This system is now distributed and the exchange of information between the COSIF and the application nodes has to be well defined.
- The isolation of the applications is complete behind the defined interface. This is clearly an advantage because the application can be optimised in its node without affecting the way in which the service is provided.
- One can see as a disadvantage the tight coupling of the application with the characteristics of the node. The application is not portable and cannot be deployed uniformly like the COSIF in other software/hardware architectures.

Taking into account the SOA aspects of the COSIF and the characteristics of the reused software described in the previous section, it is possible to define a common interface for the integration of the legacy software in the COSIF with minimum development impact both in the COSIF and in the application software. This introduces another software package for reuse which is part of the infrastructure being used by GMV to deploy flight dynamics applications.

focussuite was designed and developed by GMV with the aim of having a generic client/server infrastructure that allowed the easy integration of new application either developed according to the focussuite paradigm or inherited...
from existing legacy systems. With evolution of the technologies and the need to provide service oriented solutions this infrastructures have turned into a client/server service oriented system that can be deployed similarly as the COSIF; it is however not fully COSIF compatible but the required steps are not major. Fig 2 shows the proposed reuse of focussuite (together with the business process that defines the interaction) as common application deployment for integration in the COSIF.

![Diagram](image.png)

**Fig. 2.** Application integration in the COSIF with focussuite and associated business process

The upper part of Fig 2 represents the service node that is essentially supported by the COSIF. The focusapi (that is ready for SOA integration) and is the key element that facilitates the intended type of integration just needs to define the adaptors that communicate the exported services with the application node. The application node in this approach is unique with respect to the interface (there may be many instances identified at different IP addresses) since the protocol for all applications is the same. The integration of each application is performed directly inside the application node with respect to its in/out/auxiliary data and the management of its process by the server. This solution also provides backend administration functions that include HMI and automation capabilities; they can be also exposed through the COSIF as any other service.

The detailed configuration of the applications is maintained by the application node. Upon invocation from the COSIF the application node provides the detailed configuration to the COSIF such that this can complete the data required for the execution and provided it to the application node before executing the application. This is a very convenient implementation because the exposed services will remain completely generic from the point of view of the operations that are exported while the details are implemented in the application node and can be updated or modified in a completely transparent manner, not only for the service but also with respect to the integration of focussuite into the COSIF.

From the point of view of the application node it is most relevant to understand that everything behind the focusapi is an existing operational system that already integrates applications coming from NAPEOS. In a reuse scenario where NAPEOS is used as core flight dynamics library to build new applications the approach is readily compatible regardless whether focussuite is used or not. The approach is still valid with a different infrastructure.

In addition to the described functionalities the focusapi also implements a message and notification protocol that facilitates the communication between focussuite and the application that integrates it, in this case the COSIF. In this way it is possible to monitor and control not only the applications but the focussuite infrastructure itself.

**PROPOSED REUSE SCHEME**

With all the elements described above it is possible to define a reuse scheme that could constitute the basis for the development of SST precursor services within the SSA program. It is important to understand that the objective is to fulfil the known requirements by means of existing software that is available at ESA to maximum possible extent, being SSA an ESA program.

- **NAPEOS**, to provide general flight dynamics support in the form of a reusable library. NAPEOS is mostly data oriented (not quite object oriented) and well modularised and structured. This permits the extraction of generic supporting libraries for most low and medium level functions that can be reused for any new applications. Additionally NAPEOS provides the propagation functions that could be made reference for the SSA services as
a consistent and unique source of ephemerides. Other models like tracking reconstruction and correction, atmosphere, ionosphere, etc can also be reused on a case by case basis as they may depend on specific characteristics of the problem being solved.

- **CRASS** and **ODIN**, are not directly reusable since the software architecture is not adequate. Selected modules providing specific functionalities can be recovered but the approach would be to integrate them in the NAPEOS infrastructure described in the previous bullet. The particular case of CRASS is most interesting from the validation point of view; CRASS does not scale very well with the increasing number of objects and is not well suited for parallelisation; however, the operational robustness of CRASS makes a very valuable tool to provide reference data for the validation of new approaches to the collision risk assessment problem where the *smart sieve* algorithm is to be completely assumed as the reference solution of the problem.

- **closeap** as an evolution of CRASS can be initially reused as the solution for collision risk assessment. It implements the *smart sieve* consistently with the previous statement and is well suited for parallelisation and distributed processing allowing good scaling with an increasing number of objects. It is also integrated in NAPEOS.

- **focussuite** with reference to client/server part. This can be used for the integration of legacy application (mainly sharing the NAPEOS infrastructure) in the COSIF. At least while the application software evolves to a more service oriented design, the use of **focussuite** allows the easy integration of applications with minimum impact in the COSIF and in the applications.

**CONCLUSIONS**

The presented paper identifies a collection of software tools from the flight dynamics context that provide functionality that is common with the SST context. Obviously there are many other tools and packages that provide equivalent functionality, but the identified ones either belong already to ESA or are close enough to allow easy access to them (e.g. GMV software developed under ESA contracts).

The proposed solution for software reuse provide an initial infrastructure that guarantees that many of the core functions required for SST come from an operational, robust and well validated collection of tools and packages. This permits that the deployment of the SST precursor services can focus on the analysis of the SST specific topics with sufficient confidence that the underlying functionality minimises development and risk.

**REFERENCES**

[5] Study Note on Conjunction Event Predictions (CRASS), GMV-CRASS-SN4, 1.0, December 2001
[6] closeap White Paper, GMVINC 20042/09, 1.0, August 2009