Applying MBSE across Flight and Ground on Small and Nano-Satellite Missions

Peter Mendham
Bright Ascension Ltd, Scotland, UK
peter@brightascension.com

The so-called “New Space” industry represents a particularly challenging environment for software. This paper presents the commercial development and application of an Model-Based Software Engineering (MBSE) solution to a diverse range of small and nano-satellite missions. Particular attention is paid to the successful use of this solution across both flight and ground software, and the benefits this has brought.

The paper concludes by summarising lessons learned and looks to future developments which will expand the scope and capabilities of the solution.

Commercial Context

There is currently a global boom in the development and application of small satellites, usually characterised as those with a mass between 1kg and 50kg. In many cases, the approach taken to the development and deployment of these satellites is one that accepts a higher risk in product assurance for significantly lower development costs and faster development times.

Characteristics of missions in this category typically include:

- **rapid development times**, commonly 6-12 months per satellite and 1-2 years from conception to being able to offer an initial commercial service;
- l**arge numbers of satellites**, with constellations of 10-50 satellites being common targets although there are high-profile cases of organisations aiming to deploy hundreds of satellites;
- a **high degree of heterogeneity** between satellites in a constellation;
- **complex payloads** which embed much of their functionality in software;
- use of **Commercial Off The Shelf** (COTS) hardware from a wide range of vendors;
- the need for **highly automated operations**, typically aiming for unattended operations for the duration of a week at a time; and
- use of commercial **Ground Station Network** (GSN) providers.

This places consequent demands on the mission software, across both flight and ground.

User Needs

Space-based service developers utilising small and nano-satellites are facing a range of challenges which give rise to demands on the mission software:

- **availability**, software must be available early (at least partially) to support development and test;
- **flexibility**, requirements during development change as design is iterated;
- **rapidity**, overall schedule is short and software must be ready quickly;
- **capability**, many spacecraft functions are implemented in software;
- **operability**, software must make it easy to achieve mission and service delivery;
- **reliability**, software is mission-critical and must be robust;

- **scalability**, flight and ground software must integrate to form part of a complete system which may include multiple, complex spacecraft and ground segments.

Addressing these challenges places great demands on software, resulting in complexity both in the software itself and in the process of applying it to the mission or service.

The GenerationOne Approach

In response to these needs Bright Ascension have developed a technology called GenerationOne which combines:

- **model-based** software engineering, permitting machine comprehension of software architecture and the use of tools to assist with software development and product/quality assurance;
- **component-based** software engineering, enabling reuse of software across a wide range of scenarios and applications, combining software with its documentation and tests within libraries; and a
- **service-oriented** architecture, providing consistent and well-defined semantics for component interactions at all levels, enabling low-level aspects of the system to be expressed as components whilst improving operability.

GenerationOne technology comprises a meta-model definition, a language-independent set of service and protocol definitions, cross platform tools and framework implementations for target platforms. The GenerationOne approach permits almost the entirety of a software system to be expressed as components, from hardware drivers and communications protocols to applications. The underlying framework is lightweight and most components are portable across platforms and operating systems.

The component and service model is specifically designed to be applicable across both flight and ground environments with the model capturing both ground- and flight-based functionality. The encompassing nature of the model also extends to the life-cycle, with the model representing the system from early payload prototyping through development, Assembly Integration and Test (AIT) through to in-orbit operations.

Developing and Implementing GenerationOne

Bright Ascension’s primary focus for GenerationOne has always been to solve industry problems. As such, the technology was not fully designed up front, but instead has been developed iteratively, with improvements, additions and changes based on the experience gained from deployment in operational missions. Here the technology has benefited from the rapid launch cadence of the small and nano-satellite industry with twelve on-orbit spacecraft making use of GenerationOne, with many more in development and some slated for launch this year.

GenerationOne has been used by customers on all six continents across a wide variety of computing environments, including as flight software running on a number of COTS Onboard Computers (OBCs). The process of supporting so many target platforms has introduced a number of improvements and revisions to permit optimisation for low resources, and flexibility to different processor and operating system architectures.

Mission Case Studies

It is perhaps illustrative to highlight two missions which have benefited from the use of GenerationOne technology.

The **KIPP** and **CASE** spacecraft, intended to pilot Kepler Comminations’ Internet-of-Things (IoT) constellation are 3kg nano-satellites flying a single, highly-capable Software Defined Radio (SDR) payload. With a strong commercial case, these spacecraft were developed by AAC Clyde

Space within 8 months from concept to launch shipment. Bright Ascension used GenerationOne to develop and deliver flight and ground software in 5 months using 6 months of engineering time. The model developed from the flight software was ingested into the ground software for immediate use with minimal configuration effort necessary.

Faraday-1, the first of In-Space Space Missions' hosted payload spacecraft offers different challenges in terms of complexity and operability. Despite having a mass of less than 10kg, Faraday-1 includes six payloads from different organisations, including four SDRs each of which hosts multiple software applications, effectively "massless payloads". Faraday-1 is a highly distributed system, with a total of 6 OBCs of three different architectures and operating systems requiring 13 software images. The complete system is captured in a single model which can be viewed and used by both development tooling and the Mission Control Software.

Benefits of a Coordinated Flight-Ground Approach

The interface between the spacecraft and the ground segment sits within the software functions responsible for managing and delivering the mission and overall service. As such, although it is traditional, selecting the spacelink as the position for a significant division in system composition and even development responsibilities is a short-sighted decision leading to poor architecture and inefficient mission delivery. There are many reasons for this division spanning technical, organisational, commercial and political concerns. Within small and nano-satellite missions, most of these drivers do not exist, leaving technical arguments at the forefront.

There is no doubt that many of the challenges faced by flight and ground software are different, often generated by the distinct computing environments: from low-resource, real-time embedded systems, to enterprise systems with many simultaneous human operators handling large quantities of data. However, with GenerationOne, Bright Ascension has shown that a well designed architecture can accommodate these technical differences within one coherent system. This gives significant advantages for all aspects of system design, development, operation and maintenance.

Key to these cross-system efficiencies is the ability to capture the complete system in a single model which describes the functional architecture and its relationship with the physical system. The model acts as the basis for increased operability as well as the basis for a domain model which can be used by automation systems, including planning.

Conclusions and Future Work

The application of MBSE to Bright Ascension's GenerationOne technology has been instrumental in creating a technology which facilitates the rapid development and efficient operation of complex missions. Key to this has been the development of a single coherent architecture and meta-model which can be utilised across both flight and ground systems.

The development of GenerationOne has been incremental and is far from complete. Current improvements are focussed on scalability, operability and automation, impacting all aspects of GenerationOne from the meta-model through to standard service definitions. Once this next phase of development is complete, it is expected that the opportunity will be taken to supplement the model with a more capable range of tooling and support infrastructure, offering benefits to a wide range of stakeholders involved in mission and service delivery.