ABSTRACT

The On-Board Control Procedures (OBCP) concept is a powerful way to control spacecraft or to implement on-board functions. OBCPs are flight control procedures that can be resident on-board or that can be uploaded to the spacecraft as required by the ground. They serve for controlling processes, which may be active for an extended period of time and which may involve the execution of a sequence of commands.

This paper provides a first feedback of two parallel ESA activities that, starting from the recent ECSS standard on OBCPs [EcssObcp], make a step towards a standard OBCP building block by setting the requirements and defining its interfaces. Each of the parallel activities performs the following main tasks:

- Assessment of existing OBCP implementations
- Assessment of the ECSS OBCP standard
- Assessment of potential innovations and description of the vision of the future use of OBCPs
- OBCP building block definition, including requirements specification and interfaces
- Prototype implementation and demonstration

The two parallel activities are currently on-going and they are planned to conclude in 2012.

1. MOTIVATION AND OBJECTIVES

An ECSS standard on OBCPs has recently been published, [EcssObcp] and is providing a standard definition of an OBCP system and the development lifecycle for OBCPs and it relationships the one of the spacecraft on-board software (OBSW). The existence of the standard is a starting point on the way to the future creation of a reusable and standard flight-qualified OBCP SW system building block.

The objective of the two parallel activities is to define the OBCP building block, define its provided and required interfaces and its position within the ESA OBSW reference architecture defined in SAVOIR-FAIRE [SF]. Other objectives are to provide a vision of the future use of OBCPs, supported by a set of use cases and to consider options for further steps in the standardisation of OBCPs, such us updating the current OBCP standard or standardising the OBCP language.

2. REVIEW OF EXISTING OBCP IMPLEMENTATIONS

The available documentation for a number of different OBCP implementations has been reviewed. The level of documentation in the available documentation varies greatly. The list of missions covered includes Herschel-Planck, GOCE, Sentinel-1, Cryosat-1, Rosetta, Venus Express, BepiColombo, Columbus and the NASA VML system. Some of the aforementioned OBCP implementations are derived from each other, like in the case of Herschel-Planck that is derived from the GOCE OBCP implementation.

The definition of the OBAP and OBO concepts given in the standard [EcssObcp] is relatively recent, so existing ESA missions are only referring to OBCP.

It has to be said first that two broad classes of implementations exist:

- Command sequences: the OBCP is made of a sequence of telecommands on a relative time-line with a limited (if any) execution logic, similar to ground operations, but permanently resident on-board. This class is more suitable for the implementation of fixed operations sequences.
• Script-like: the OBCP is written using a high-level language, possibly compiled, and executed on-board by an on-board interpreter. This class is more suitable to implement complex control logic but requires a more complex OBCP engine than the previous case.

The implementations have been analysed and compared in their main aspects:
• Language capabilities
• OBCP preparation environment capabilities
• OBCP execution environment capabilities
• OBCP management capabilities

3. ASSESSMENT OF THE ECSS OBCP STANDARD

The [EcssObcp] standard has been reviewed and a number of comments have been given, for potential future improvement.

The standard introduces a number of concepts like OBAP (On-Board Application Procedures) vs. OBOP (On-Board Operations Procedures) and covers all the main aspects of OBCPs, including
• Use cases
• Language capabilities
• Preparation environment capabilities
• Execution environment capabilities
• Engineering process, taking into account the distinction of OBOPs vs OBAPs

The assessment of the current ECSS standard shall serve for considerations of further steps of standardising OBCPs, but also to propose a possible update of the current standard, improve understanding of the standard and trigger a discussion leading to the definition of the OBCP building block.

The main areas identified for possible improvement of the standard are:
• Standardisation of the provided and required interfaces of the OBCP execution environment with respect to the hosting system, ideally the ESA OBSW Reference Architecture, see also chapter 5 of this paper.
• Prioritisation of the requirements taking into account the needs of the various users and the goal of integration with the ESA OBSW Reference Architecture.

4. ASSESSMENT OF POTENTIAL INNOVATIONS

The assessment of potential innovations has started from taking the “best of” from the various implementations, proposing them as baseline for the future standard OBAP OBSW BB and has introduced a number of additional concepts like:
• Explicit support of state machines,
• Standardisation of the target “object code” language,
• Source language improvement to better support the users, e.g. support of at least basic forms of object-orientation
• Standardised interface towards the system database
• Integration with the SAVOIR-FAIRE standard on-board SW architecture

The potential innovations are driven by user requirements and considerations for future use of the OBCPs in flight software.

Autonomy aspects from past missions and the mechanisms used to achieve autonomy in spacecrafts are also analysed. Future autonomy will even go further to “level E3, adaptive autonomy” [ECSS-E-70-11], where the spacecraft is able to adapt its behaviour to new situations. Autonomy is the main driver for developments of OBCP in the near future.

The studies has also evaluated some of the most promising solutions concerning the most influential technologies of current flight software, such as Time and Space Partitioning [TSP], SOIS architecture [SOIS], Component-oriented frameworks [CBSE] and the relation to FDIR (Fault Detection Isolation and Recovery) functions.

5. REQUIREMENTS AND INTERFACES OF THE OBCP BUILDING BLOCK

The activities shall produce a set of system requirements for the OBAP Building Block. The requirements should be compliant with the OBCP ECSS standard and should be accompanied with use cases to demonstrate their purpose.

The ESA OBSW reference architecture is one of the inputs for this task, together with the review of current OBCP implementations. The objective is to define the OBCP building block in terms of its provided and required interfaces and interaction to flight software and spacecraft operations.

6. OBCP BUILDING BLOCK PROTOTYPE

Based on the previous tasks, the two studies have made two different prototype implementations in order to demonstrate the feasibility of different concepts.

The current development is summarised in the chapters below.

6.1 SSF EXPERIENCE

The prototype implementation for the SSF project is ongoing at the time of writing this paper. Starting from the very advanced OBCP system used for the Herschel-Planck missions it focuses on:
• Implementation and demonstrations of new concepts like:
  • Explicit support for state machines to facilitate OBCP design and implementation
6.2 GMV EXPERIENCE

At present, the specification of requirements and interfaces is progressing with a focus on core requirements for OBCP-BB, considering current ECSS-E-ST-70-11 standard specification.

The high level architecture that is proposed is in line with the COrDeT OBSW Reference architecture, [COrDeT].

7. CONCLUSIONS

Starting from the assessment of past OBCP implementations and of the ECSS OBCP standard, the activity has proposed a number of improvement and innovations; it has set the requirements and interfaces for a future standard on-board SW OBCP building block and has provided two prototype implementations for concept demonstration.

Inputs have been provided for the OBSW reference architecture interfaces and for the improvement of the OBCP standard.

As future work it is envisaged the implementation of a pre-qualified building block, to be made available for future space missions.

8. ACRONYMS

BB Building Block
CBSE Component-Based Software Engineering
COrDeT Component Oriented Development Techniques
FDIR Fault Detection, Isolation and Recovery
OBSW Flight Software
IMA Integrated Modular Avionics
OBAP On-board Application Procedure
OBCP On-board Control Procedures
OBOP On-board Operations Procedure
OBSW On-Board Software
PUS Packet Utilization Standard, ECSS-E-ST-70-01C
SOIS Spacecraft On-board Interface Services
TSP Time and Space Partitioning

9. ACKNOWLEDGEMENTS

The two activities are supported by the OBCP-BB ESA General Support Technology Programme (GSTP) study (“Requirements and Interface Definition for Future OBCP Building Block”);

The SSF study is conducted in collaboration with the Technical University of Braunschweig.

The study lead by GMV involves Space Applications (SA) and Evolving Systems Consulting (ESC).

10. REFERENCES


[Steiger2005] Christoph Steiger, Robert Furnell, Jose Morales: On-Board Control Procedures For ESA’s Deep Space Missions Rosetta And Venus Express, DASIA 2005, Edinburgh, UK, 30 May – 2 June 2005


[VML2008] C. Grasso, P. Lock “VML Sequencing: Growing Capabilities over Multiple Missions”, American Institute of Aeronautics and Astronautics


[COrDeT] COrDeT-2 ESA TRP study; Webpage: http://COrDeT.gmv.com/