The European Ground Systems Common Core (EGS-CC) Initiative

Anthony Walsh[1], Mauro Pecchioli[2],
European Space Agency, ESA/ESOC, Robert Bosch Strasse 5, 64293 Darmstadt, Germany

Juan María Carranza [3]
European Space Agency, ESA/ESTEC, Keplerlaan 1, 2201 AZ Noordwijk ZH, The Netherlands

Wolfgang Bothmer [4]
OHB System, Universitätsallee 27-29, 28359 Bremen, Germany

Pierre-Yves Schmerber [5]
Thales Alenia Space France, 100 boulevard du Midi, BP 99 - 06156 Cannes, France

Johannes Rueting [6]
EADS Astrium Space Transportation, P.O.Box 28 61 56, 28361 Bremen, Germany

Pascal Parmentier [7]
EADS Astrium Satellites, 31 rue des Cosmonautes, Z.I. du Palays, 31402 Toulouse Cedex 4, France

Paolo Chiroli [8]
Thales Alenia Space Italy, S.S. Padana Superiore, 290, 20090 Vimodrone (MI) - Italia

Marie-Claire Charmeau [9]
CNES, Centre spatial de Toulouse, 18 avenue Edouard Belin, 31401 Toulouse Cedex 4, France

Michael Geyer [10]
DLR, Münchner Straße 20, 82234 Weßling, Germany

The European Ground Systems – Common Core (EGS-CC) is a European initiative to develop a common infrastructure to support space systems monitoring and control in pre- and post-launch phases for all mission types. This is expected to bring a number of benefits, such as the seamless transition from spacecraft Assembly, Integration and Testing (AIT) to mission operations, reduce cost and risk, support the modernisation of legacy systems and promote the exchange of ancillary implementations across organizations. The initiative is being performed as a collaboration between ESA, European National Agencies and European Industry. In this paper we describe the main objectives of the EGS-CC initiative and the overall system concept and features it will provide. The current status and future planning of the EGS-CC programme are also reported.

[1] EGS-CC system engineering team member, anthony.walsh@esa.int
[2] EGS-CC system engineering team chair, mauro.pecchioli@esa.int
[3] EGS-CC system engineering team deputy chair , juan.carranza@esa.int
[4] EGS-CC system engineering team member, wbothmer@ohb-system.de
[5] EGS-CC system engineering team member, pierre-yves.schmerber@thalesaleniaspace.com
[6] EGS-CC system engineering team member, johannes.rueting@astrium.eads.net
[7] EGS-CC system engineering team member, pascal.parmentier@frsat.astrium.eads.net
[8] EGS-CC system engineering team member, paolo.chiroli@thalesaleniaspace.com
[9] EGS-CC system engineering team member, marie-claire.charmeau@cnes.fr
[10] EGS-CC system engineering team member, michael.geyer@dlr.de

I. Introduction

A. Background

Within Europe there are many different systems for monitoring and control used by companies/agencies for space system operations and Assembly Integration and Testing (AIT). Some of these systems are common to both operations and AIT, while some are specific. Often multiple systems are used in AIT of a space system by different companies or at different levels (e.g. payload/system) or in different phases. Many of these existing systems have reached or are reaching their end of life. The systems are often using old software technologies and hardware platforms that are difficult to modernise. The maintenance and evolution costs are therefore becoming excessively complex with time. The compatibility/exchange of information with other systems is also often difficult leading to little synergy across missions and project phases

Given the difficulties mentioned above, during 2009-2010, the European Space Agency (ESA) discussed with large European System Integrators, including Astrium Satellites, Astrium Space Transportation, Thales Alenia Space (France and Italy) and OHB System, the possibility of a collaboration to develop a European Ground Systems Common Core (EGS-CC) which would provide a common infrastructure to support space systems monitoring and control in pre- and post-launch phases. The French and German national space agencies, CNES and DLR also signalled their desire to join the initiative and a Memorandum of Understanding was finalised in support of the EGS-CC initiative.

B. Objectives

The objective of the European Ground Systems – Common Core (EGS-CC) is to develop a common infrastructure to support space systems monitoring and control in pre- and post-launch phases for all mission types. This is expected to bring a number of benefits, including:

- The seamless transition from spacecraft Assembly, Integration and Testing (AIT) to mission operations, thus maximising synergy across all mission phases
- Enable overall cost reductions by sharing development, sustaining and maintenance of a single infrastructure across organisations
- Facilitate cost and risk reduction when implementing space projects through the provision of a stable common infrastructure which can be easily tailored for the needs of a specific mission and/or organisation
- Enable the modernization of legacy EGSE and MCS systems
- Enable the exchange of ancillary implementations across organizations

The objectives of the EGS-CC are therefore very ambitious and the main system features include:

- Support of all mission types and phases
- Open, component based, service oriented architecture
- Generic and extensible functionality
- Binary compatibility
- Layered implementation
- Clear separation between generic M&C functions (kernel) and specific features of the controlled system (adaptation layer)
- Configurable level of operations abstraction
- Standardised interfaces (as far as possible…)
- Technology isolation (as far as possible…)
- Long term maintainability
- High performance and scalability

II. System Concept

A. Monitoring and Control Model

The Monitoring and Control Model (MCM) provides the functional core of the EGS-CC kernel. It includes the capability to model the complete space system from a monitoring and control perspective. It encapsulates the main monitoring and control functions (e.g. parameter processor, activities handler, events processor) and provides access to all data of Monitoring and Control (M&C) relevance, thus acting as an abstraction layer for M&C operations. It is based on the principles of the M&C view of the Space System Model (SSM) defined within the ECSS E-ST-70-31C Ground Systems and Operations – Monitoring and Control Data Definition [4]. Information related to the space system is organised in a structured way which reflects the functional decomposition of the space system, including the controlled system as well as the control system itself and covering both the Space Segment (e.g. a satellite or launcher) and Ground Segment (i.e. the overall infrastructure required during Development, Assembly, Integration and Testing and for Operations) for the AIT or operational contexts. The decomposition of a typical space system as depicted by the MCM is illustrated in the Figure 1.

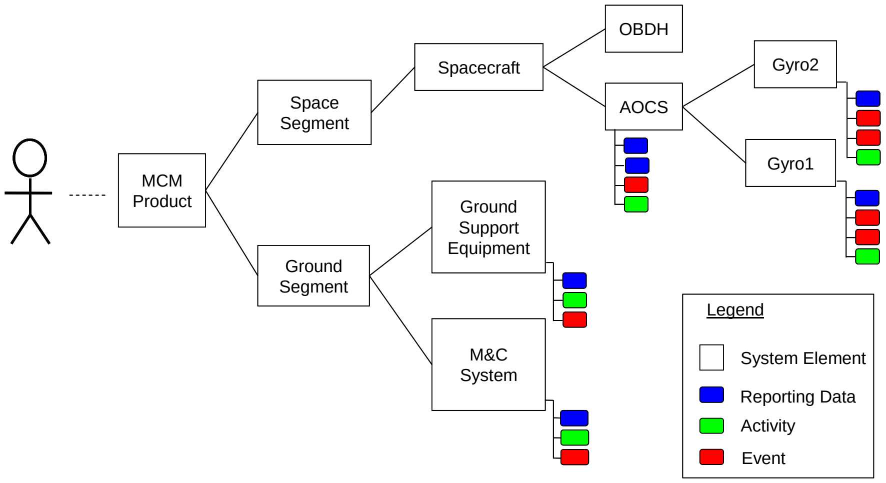


Figure 1. Monitoring and Control Model

The Figure 1 shows the space system is represented in the MCM as a hierarchy of System Elements (SE). The system element is a data structure whose properties provide the means to organise the space system knowledge in a hierarchical structure. From the highest level downwards, this is typically: system, subsystem, set, equipment or software product, assembly, part (hardware) or module (software), but it is not required to be a one-to-one mapping of the physical decomposition of the space system. The space system monitoring and control knowledge is categorised into activities, events and reporting data, according to the following definitions:

- Activity – An activity is a space system monitoring and control function supported by the EGSE or Mission Control System (MCS). An activity can be executed within the EGS-CC based system (e.g. a procedure, a script, a call to a system function) or implemented as a command either to the space segment or any other command type that is specific to a given implementation of the space system (e.g. a command to a Special Check-Out Equipment (SCOE) or to a ground station using a proprietary protocol).
- Event – An event is an occurrence of a condition or group of conditions of operational significance. Events are widely used within the space system to trigger the execution of functions (e.g. acquisition of signal can initiate telemetry processing tasks at the ground station).
- Reporting Data – Reporting data is information that a system element provides, irrespective of how this information is used. Reporting data can comprise measurements which reflect the state of the associated

system element or an output product whose purpose is to be used by another system element (e.g. manoeuvre parameters provided by a flight dynamics system).

Although the MCM organises reporting data, activities and events into a more logical structure, the view presented to the end user can be customised as required. For example, it is still possible for the end user to view and interact with the system as a more traditional flat list of parameters and commands if so desired.

The MCM concept ensures a clean separation between the M&C abstract view and generic processing and the specific processing related to the data units exchanged with the controlled system. This approach allows the application of the same M&C kernel to different types of controlled systems, such as spacecraft, EGSE systems, ground station equipment, etc. It means that the M&C system can control not only the target system (e.g. a spacecraft), but also all other contributing ground systems (e.g. EGSE supporting equipment, and the EGS-CC itself), with which the exchanged data are not necessarily based on TM/TC packets. Knowledge about the Space System can be "static" or "dynamic" in nature.

- The MCM static knowledge encompasses all monitoring and control data produced during the development and the maintenance of the space system and is used to tailor an EGS-CC system for a given mission.
- The dynamic knowledge encompasses all monitoring and control data produced by the space system during testing and operations (e.g. the archive of all sent telecommands and all received telemetry, the output of the ground processing of monitoring data). In EGS-CC, this is referred to as the MCM State

The relationship between the MCM static knowledge and the MCM state is illustrated in the Figure 2. The MCM static knowledge is stored in the M&C database while the dynamically generated MCM state information is stored in a data archive.

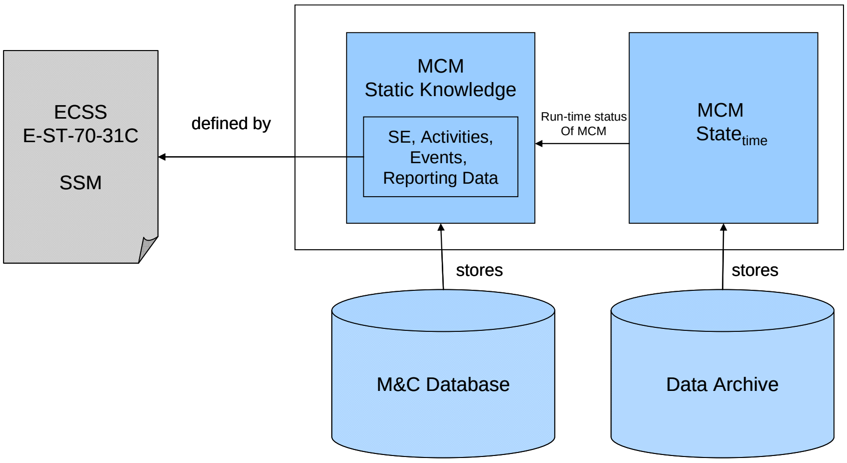


Figure 2. MCM Static and Dynamic Knowledge

B. Processing Model and Data Archiving

The MCM is a model of the complete space system for the purpose of monitoring and control. The main inputs to the MCM therefore is monitoring data received from the controlled system, referred to as raw monitoring data. There is also other data generated by the EGS-CC applications inside and outside the kernel which can change the state of the MCM (e.g. requests to initiate activities). Both the raw data and the internally generated source data are considered source data and are stored in a source data archive. The source data is fed into an MCM processing model which generates the dynamic state of the MCM which is stored in the processed data archive. This is illustrated in the following Figure 3.

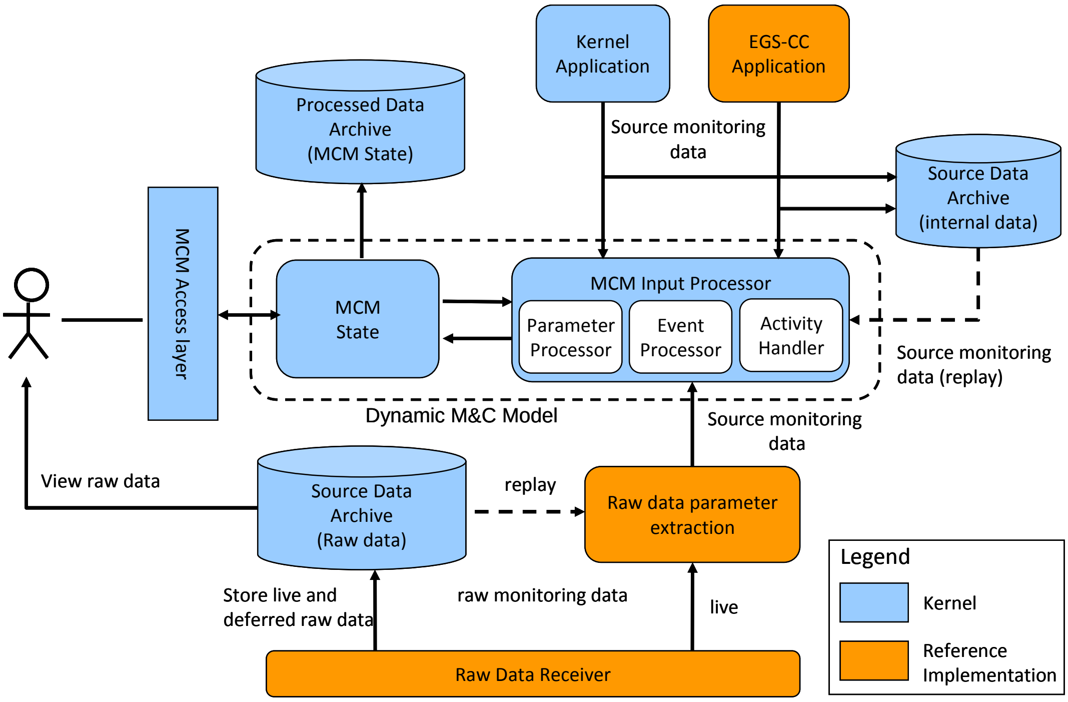


Figure 3. Data Processing

The processed data archive is populated with data from the MCM processor when raw data is received from the controlled system in live mode or when the raw and internal data stored in the source data archive is re-processed in replay or reprocessing mode. The processed data may be retrieved from the processed data archive at any stage and does not require the processing of data stored in the source data archive.

C. System Organisation

The EGS-CC system organisation is based on the concepts illustrated in the following Figure 4.

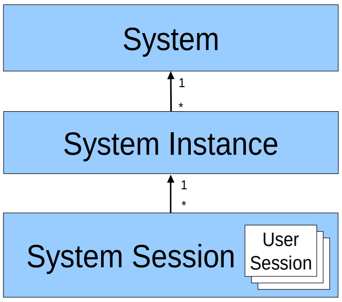


Figure 4. System Organization

The EGS-CC system comprises of the software source code and binaries for a given EGS-CC based system, applied to a specific application and mission. The system is composed of at-least the EGS-CC Kernel which is independent of a specific mission, specific adaptations and any other additional auxiliary systems. A system instance is a particular system release installed on a specific target platform. The running system instance is the set of processes executed at run-time, which belong to a single system instance, which can be managed together in terms of start/stop and status monitoring.

The system session is a logical grouping of EGS-CC components belonging to a given running system instance, which use a given tailoring and configuration , dedicated to the processing of the data of a controlled space system and generating a separate set of outputs within the processed data archive. In the context of EGS-CC, tailoring is process of mission data definition for adapting customised EGS-CC system software (if needed) to a given mission and operations concept, while configuration is the tuning of the configuration parameters for the run-time system (i.e. without the need to recompile the system).

There may be different types of system sessions: operational, simulations, and tests. Several system sessions can run in parallel, especially in the space AIT environment, producing independent set of results. Within a specific system session, there may be any number of user sessions which provide the execution of a specific context dedicated to a given user.

D. Role of Standards

The EGS-CC is intended to support a number of standards which will enable the EGS-CC to interoperate with other systems and other segments. The open nature of the EGS-CC allows support for standards to be added as specific adaptations, but a number of standards will be supported as part of the base implementation, including:

- Telemetry and telecommand protocols (ECSS-E-ST-50-03 and 04) [1], [2]
- M&C data definitions (ECSS-E-ST-70-31) [4]
- Procedures languages (ECSS-E-ST-70-32) [5] although this does not mean that all procedures languages supported by EGS-CC must meet these requirements but rather that it must allow languages compliant with this standard (e.g. PLUTO)
- Packet Utilisation Standard (PUS) (ECSS-E-ST-50-41) [6] for missions implementing those services
- Standards related to services to be used or offered: MAL (CCSDS 521.0-B) [8], Mission Operation Services Concept (CCSDS 520.0-G-3) [7] and SLE (CCSDS 9xx) [9]
- On-board control procedures (ECSS-E-ST-70-01) [3]

III. Conceptual Architecture

A. Architectural Layers

The EGS-CC architecture is logically structured in separate layers, isolating different responsibilities and defining clear interfaces between them. The following Figure 5 shows the architectural layers of the EGS-CC.

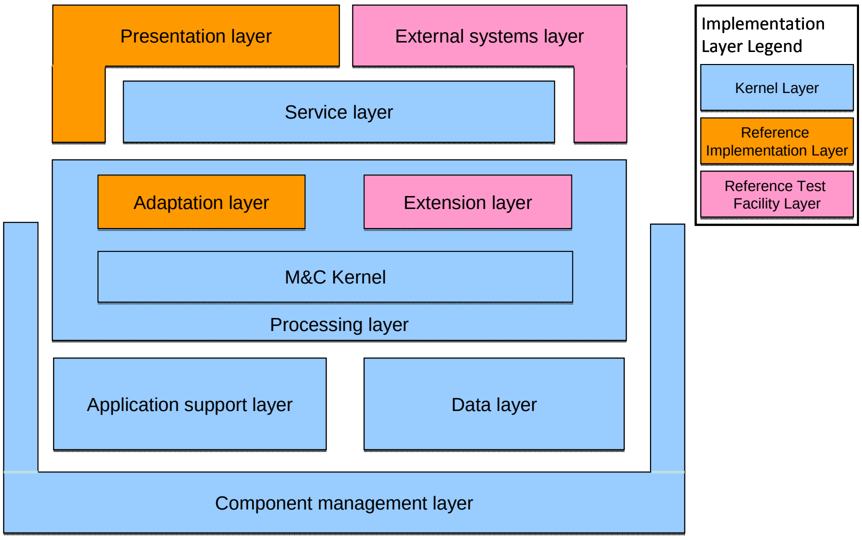

Presentation layer
External systems layer
Service layer
Adaptation layer
Extension layer
M&C Kernel
Processing layer
Application support layer
Data layer
Component management layer
Implementation Layer Legend
Kernel Layer
Reference Implementation Layer
Reference Test Facility Layer

Figure 5. Architectural Layers

The lower architectural layers provide cross cutting features used across all EGS-CC functions and include:

- Component management, providing the framework in which the EGS-CC components are developed and executed. It is the lowest level platform of the application, in which the application support and processing components are integrated.
- Application support, providing general purpose components and services (e.g. logging) to ease the development of processing or application components in a harmonised way.
- Data layer, responsible for the storage, distribution and management of data.

The processing layer contains the business logic of the system. It includes all the components which model the processing of the system. It contains the following important sub-layers, according to their functionality:

- M&C component layer, providing generic M&C components which implement the common M&C functionality for the different applications (e.g. procedure execution, scripting execution, generic processing models, etc,).
- Adaptation layer, containing the components which implement application and mission specific functionality, to adapt the EGS-CC to different target applications and controlled systems. The adaptation layer includes specific interfaces with external systems, such as SLE transfer with ground stations for spacecraft missions, or applicable protocols to interact with ground station equipment, SCOEs, etc.
- Extension layer, containing components with additional functionality specific for different applications or missions, which extends the standard monitoring and control functionality provided by the generic and adaptation layers.

The service layer provides access to the lower layers, encapsulating internal interfaces into well-defined services. It provides the provision of services of EGS-CC components to external consumers (giving access to internally processed data). The presentation layer contains user interface components, which will be mainly dependent on the different applications and missions. They may interact with the processing layer directly, or through the service layer. The external systems layer represents the external systems, or their emulated versions implemented for testing.

The EGS-CC functionality is also organised in different layers for its implementation, depending on their purpose and scope and include:

- Kernel, which contains the generic functionality applicable to all applications and missions, for both general purpose, and monitoring and control functions. It is intended to be reused across all EGS-CC based systems for all applications, missions and purposes and cannot be modified.
- Reference implementations, containing application specific functionality, intended to be applied in some but not all EGS-CC based systems. This layer is what adapts EGS-CC software to a given application and mission. The reference implementations may be replaced or adapted for the specific needs of a mission or organisation. The EGS-CC will provide specific reference implementations for a selected category of missions and applications (e.g. PUS spacecraft, SLE ground stations).
- Reference test facilities, providing external functionality needed for testing EGS-CC based systems in isolation. Test facilities are not part of EGS-CC based systems, but implemented for supporting tests.

B. Architectural Styles

In order to fulfil its objectives, the EGS-CC design follows both a Component Based Architecture (CBA) and Service Orientated Architecture (SOA). This is supported by underlying middleware platforms which comprise:

- Component Framework, supporting the development of EGS-CC implementation components, their composition into EGS-CC composite components, and their deployment and execution within a component run-time platform.
- Service Integration Platform, supporting the integration of a complete EGS-CC based system through the provision of an Enterprise Service Bus (ESB).

The component framework supports the development and execution of EGS-CC components, the building blocks for a complete EGS-CC based system, while the service integration platform supports the integration of EGS-CC components, component composites and external systems to form a complete EGS-CC based system. The service integration platform is able to integrate heterogeneous systems communicating via different protocols, while the component framework offers a harmonised and optimised approach to component implementation based on a specific component model.

The following Figure 6 illustrates the run-time context of the component framework and the service integration platform within the EGS-CC architecture. The top level composite shown in the figure comprises of two lower level component composites, containing implementation components. The contained implementation components can be either part of the kernel, reference implementation or reference test facility implementation layers introduced earlier. The example also illustrates the required and provided interfaces of components which can promoted to the level of the composite in which the component is contained. The composite also defines the connections between the provided and required interfaces of its internal components.

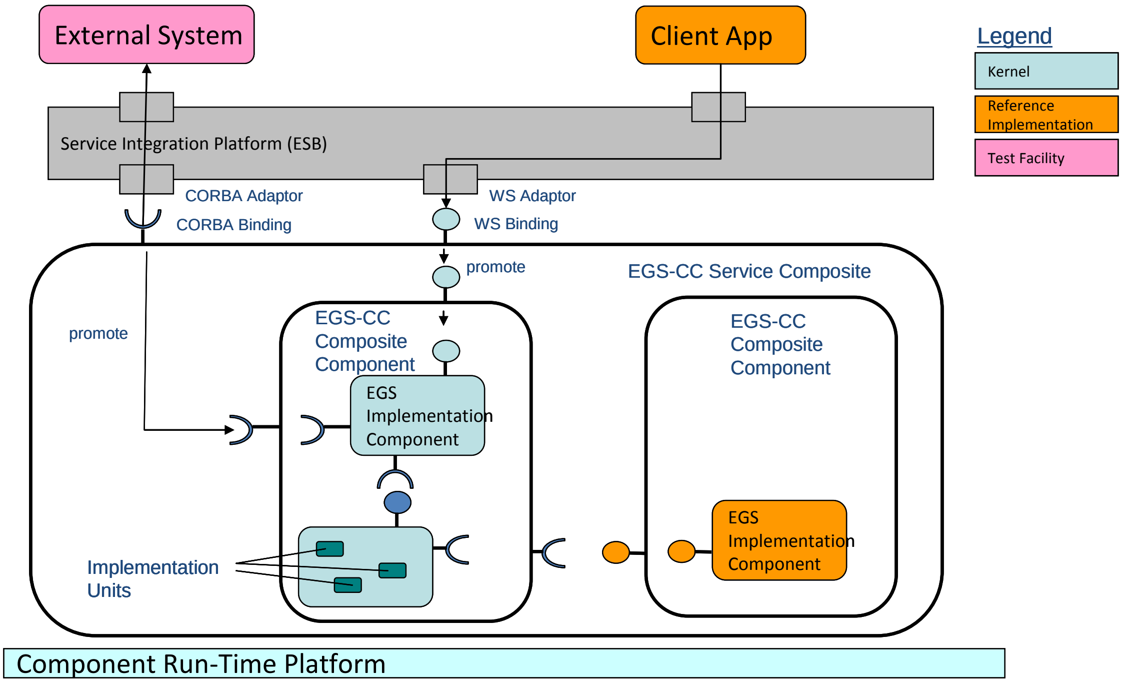


Figure 6. System Integration

In the figure 6, the top level composite, called a service composite, is deployed within the component run-time platform. The interfaces promoted to the top level composite may be integrated with other EGS-CC service composites or external systems. This is achieved by binding the composite’s required or provided interfaces to a specific service technology. In this example, a CORBA and a web-service binding are selected for the two interfaces of the top level composite. The service integration platform is then used to integrate these two services with an EGS-CC client application or a legacy system. In this case, the client application could represent an EGS-CC user application implementing the monitoring and control user interface. This overall approach offers great flexibility for system integration allowing many decisions to be deferred to deployment time.

C. Conceptual Data Model

The EGS-CC includes a Conceptual Data Model (CDM) which defines the data needed to operate an EGS-CC based system. This is required to improve the efficient hand-over of data (and in that engineering knowledge) on the space system throughout the project phases of engineering, development, verification and operation of a space system. The following figure 7 illustrates this concept, showing that conceptual data model is a key enabler for the exchange of data between systems, in this case between an EGS-CC based checkout system used for spacecraft AIT and the mission control system used for spacecraft operations.

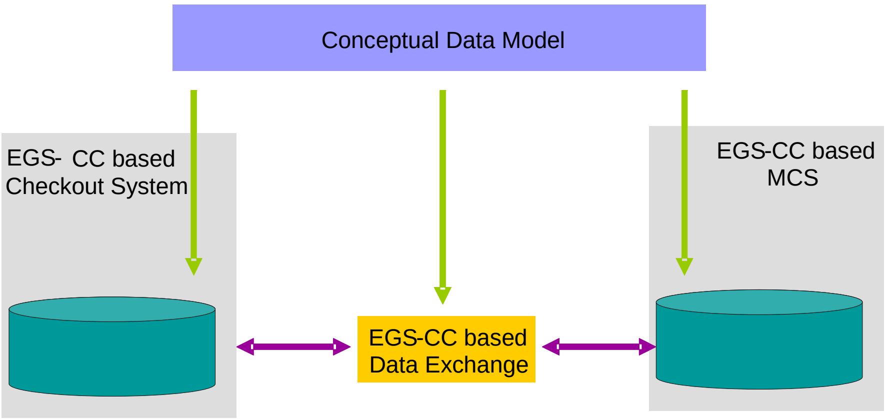
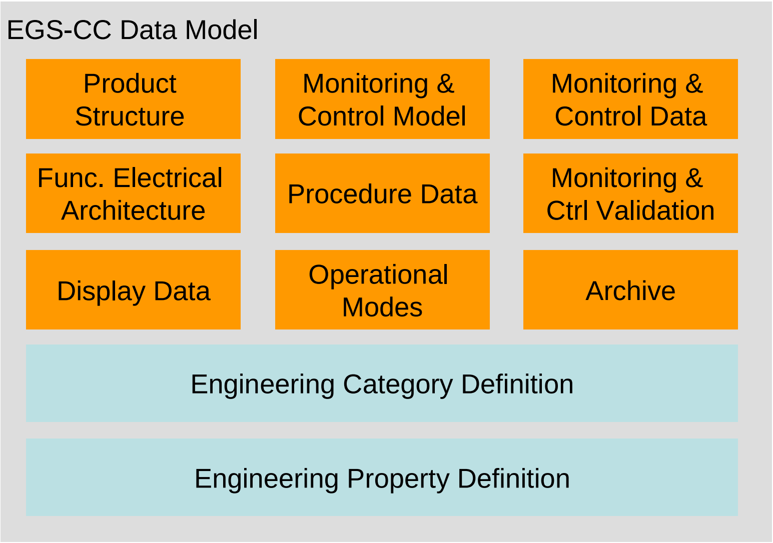


Figure 7. EGS-CC Data Exchange

The CDM identifies uniquely all data items in use along the process such that the interface between different EGS-CC based systems can be specified. The actual technology for the data exchange (e.g. XML) is not reflected in the CDM, but is such that the target representation can be generated. The CDM is therefore robust against future technology changes. The figure 8 shows the top level decomposition of the CDM and covers the data which eases the collaboration between the different parties involved in the space system engineering, development, AIT, verification and operation process.



EGS-CC Data Model

Product Structure	Monitoring & Control Model	Monitoring & Control Data
Func. Electrical Architecture	Procedure Data	Monitoring & Ctrl Validation
Display Data	Operational Modes	Archive

Engineering Category Definition

Engineering Property Definition

Figure 8. Conceptual Data Mode Decomposition

D. Technology Selection

The technologies to be selected for the implementation of the EGS-CC are currently under investigation. The Linux operating system has already been selected for all implementations with compatibility for Microsoft Windows also required for the user interface and preparation environment. Java has been selected as the baseline programming language.

The different technology domains where third party technologies could be applied have already been identified and selection criteria established. The following Table 1 presents the different domains and a preliminary list of technology candidates that are being considered within each domain.

Technology Domain	Preliminary list of Technology Candidates
Component Framework	OSGi, Spring, EJB3, CCM, SCA
Service Integration Platform	ESBs (OpenESB, ServiceMix, Mule, Jboss), CCSDS MAL+COM
Communication and Data Distribution	JMS (AciveMQ, Joran), AMQO, OMG Notification Service, OMG Data Distribution Service
System Run-Time Management	JMX
Logging and Tracing	Log4j, OSGi Log Service, OMG Logging Service
File Management	Revision Control: SVN, GIT, Content Repository API for Java (JCR)
Security	JAAS, LDAP
Technology Domain	Candidates
Data Persistence	Service Data Objects, Java Data Objects, Java Persistence Api, Java DataBase Connectivity, Hibernate
Data Archiving	RDBS, HDF5, noSQL (Cassandra),
Data Modelling and Editing	UML, Eclipse EMF
Post Processing and Reporting	Eclipse BIRT, JasperReports
Scripting and Automation Languages	Javascipt, Python, Groovy, PLUTO, ..
User Interface	Thick Clients (Eclipe RCP, Swing, QT), Thin Clients (Ajax, RAP,…), Flex
User Defined Displays	SVG, X3D, ..

Table 1. EGS-CC Candidates Technologies

The final selection of technology candidates will consider the best integrated technology stack across all domains. It is also important that the technology is decoupled as far as possible from the functional aspects of the software component implementations, such that it is possible to migrate to other technologies in the future with minimal impact. This is particularly important since the life span of the EGS-CC system is expected to be long and technologies will evolve significantly during this time.

IV. Programmatics

The development and maintenance of the EGS-CC is split into a number of phases, as summarised below:

- Phase A - User Requirements & System Concept
- Phase B - Software Requirements Engineering and Architectural Design
- Phase C/D - Detailed Design and Coding
- Phase E - Maintenance and Evolution

The phase A activities are currently being performed by a System Engineering Team comprising of ESA, CNES, DLR and the large European System Integrators and under the supervision of an EGS-CC Steering Board. This phase will be completed in Q3 2012 with the system requirements review which will lead to the finalisation of the EGS-CC user requirements. The phase B activities are planned to start at the end of 2012 and will be carried out by an industrial consortium, which will include MCS and EGSE developers under the supervision of the EGS-CC Steering Board and Systems Engineering Team.

The decision on the funding for the phases C/D/E of the EGS-CC programme is expected to take place in the course of 2013. Assuming a positive outcome, the detailed design and implementation activities can proceed after the completion of phase B, including an end-to-end integration in the target environments and validation for a pilot project based on the development of a full EGSE/MCS system using EGS-CC.

The phases described above depict a classical waterfall approach for systems definition and development. However, the agreed approach is to adopt an iterative approach within each phase, such to promote the continuous collaboration between the various stakeholders and the progressive consolidation of the expected deliverables taking all relevant aspects into account.

V. Conclusions

The initial phase of the EGS-CC is currently in progress and requirements engineering process defining the user needs analysis, functional and non-functional requirements is ongoing. A system concept covering usage scenarios, conceptual architecture and the associated data models is also being developed in parallel. Different technological options for the development are being identified and evaluated. The outputs of this initial phase will be completed with a wide System Requirements Review which will ensure completeness and correctness of the outputs against the needs of future missions. Once completed, the software requirements engineering and architectural design activities will be initiated, followed by the system implementation, concluding with an end-to-end validation based on the development of a full EGSE/MCS system using the EGS-CC.

The development of the EGS-CC is very ambitious and requires overcoming a number of risks. Technical consensus needs to be achieved while also synchronizing with the different stakeholders own roadmaps and schedules. This must also be done is parallel to the development of systems which will use/interface with the EGS-CC. The commitment of the stakeholders is however very strong and the objectives while ambitious are considered feasible. The expected benefits therefore justify this commitment and associated investments.

Appendix A

Acronym List

AIT	Assembly, Integration and Testing
API	Application Programming Interface
CBA	Component Based Architecture
CDM	Conceptual Data Model
CNES	Le centre national d’études spatiales
DLR	Deutsches zentrum für Luft- und Raumfahrt
EGS-CC	European Ground Systems Common Core
EGSE	Electrical Ground Support Equipment
ESA	European Space Agency
ESOC	European Space Operation Centre
EGOS	ESA Ground Operation System
EMCS	EGSE or Mission Control System
ESB	Enterprise Service Bus
M&C	Monitoring and Control
MCM	Monitoring and Control Model
MCS	Mission Control System
PUS	Packet Utilisation Standard
SCOE	Special Check-Out Equipment
SOA	Service Oriented Architecture
SE	System Element
TM	Telemetry
TC	Telecommand

References

[1] ECSS-E-ST-50-03C (Space engineering – Space data links – Telemetry transfer frame protocol), 31 July 2008

[2] ECSS-E-ST-50-04C (Space engineering – Space data links – Telecommand protocols synchronisation and channel coding), 31 July 2008

[3] ECSS-E-ST-70-01C (Space engineering – Spacecraft on-board control procedures), 16 April 2010

[4] ECSS-E-ST-70-31C (Space engineering – Ground systems and operations – Monitoring and control data definition), 31 July 2008

[5] ECSS-E-ST-70-32C (Space engineering – Test and operations procedure language), 31 July 2008

[6] ECSS-E-ST-70-41A (Space engineering – Ground systems and operations – Telemetry and telecommand packet utilisation), 30 January 2003

[7] CCSDS 520.0-G-3 (Mission operations services concept), December 2010

[8] CCSDS 521.0-B-1 (Missions operations – Message Abstraction Layer), 1 October 2010

[9] CCSDS 910.4-B-2 (Cross support reference model – Part1: space link extension services), 2 October 2005