ISI S-SOL mock-up : Key Features Induced by the Adoption of New Standards

Pierre Bornuat
(CS Communications & Systems)
March 1st, 2011

Authors: Pierre Bornuat, Pierre-Alban Cros, Christophe Pipo, CS Communications & Systems; Marie-Laure Anadon, Paul Gelie, CNES
CNES plans to develop a new control centre, which should satisfy several main objectives:

- Be compliant with international space standards (such as CCSDS: Mission Operations, Space Packet Protocol, SLE - or ECSS: PUS, etc.)
- Reduce possession costs and secure development of CNES projects
- Define an evolutive and re-usable platform product line for control centers
- Perform a first implementation to be operational in 2016

As a preparation of future developments, CNES has delegated to CS Communications & Systems the development of a mock-up (ISIS-SOL):

- Evaluate service-oriented architecture (SOA), based on the CCSDS MO standard
- Evaluate possible technologies, check real-time performances on critical subsystems

This presentation gives an introduction to the main characteristics of this mock-up, in terms of:

- Service orientation (MO standard)
- Module orientation (OSGi technology)
Service Oriented Architecture: definition of architectural layers, use of modular components, communication through service interfaces

Framework of standard services: components assembled as « plug-ins »

Framework to define new services based on the Common Object Model (COM)
ISIS-SOL mock-up Service Oriented Architecture

MO services implemented in the mock-up

**CORE Services**

- **Parameter Service**: manage Parameter status information

  - ISIS-SOL mock-up specificity: receive / distribute raw TM packets

- **Action Service**: invoke actions (i.e. remote commands)

- **Aggregation Service**: group several Parameters in a single request

  - Not used: Alert Service (raise and receive alerts), Check Service (check Parameters against monitoring rules), Statistic Service (statistical evaluation of parameters), Conversion Service (engineering unit conversion), Argument Service (access to Actions / Alerts arguments definitions)

**Common Services**

- **Directory Service**: providers publish services / consumers discover available services

- **Retrieval Service**: retrieve blocks of historical data

- **Replay Service**: replay session (used in case of investigations)

  - Not used: Login Service (authentication), Configuration Service (configuration delegation to a service), Interaction Service (request input from operator)
Modularity of telemetry acquisition chain results in a very adaptable system.

Changing On-Board telemetry format may impact a limited number of components: Demultiplexing Daemon and removal of service provider for Large Data Transfer Daemon.

Adding a new specific processing for a type of telemetry packet can be achieved by adding a new Daemon which is a consumer of a Parameter Service provided by Demultiplexing Daemon.

Using specific packet processing for each type of packets (or group of type) can be done by only changing system configuration and launching as many Telemetry daemons as necessary (for example, one per packet provider).
**ISIS-SOL mock-up Service Oriented Architecture**

**Focus on Command Control - Commands**

- Command Action is also encoded in compliance with ECSS PUS standard
- Actions states: published by the Command Control subsystem, archived by the Datastore subsystem
- Allows to follow action uploading in real-time or to replay sending command stacks
- Examples of possible adaptations to specific behaviors with a low impact on existing components, due to ISIS-SOL mock-up service oriented architecture:
  - PUS Large Commands handled by a specific component which splits them into smaller actions, and requests Commanding Daemon to take them into account (transparent to the user)
  - Handling telemetry reports for actions processing: this could be achieved by simply modifying the Commanding Daemon and register it as consumer of a Telemetry Aggregation Provider
One of the main benefits of using a MOM system for all exchanges in the system is that it provides several possible levels of Quality of Service (as required by CCSDS MO standard)

**BEST EFFORT**
- Messages are not persistent
- Sending order is FIFO
- Fault tolerant to network errors: message re-emitted until sending is successful

**ASSURED**
- Messages are persistent: a consumer will receive messages later if not present when the message was emitted
- Fault tolerant to both network and MAL (Joram) server errors. Possibly some messages loss if an error occurs on client reception

**QUEUED**
- Same as ASSURED but all messages are acknowledged on the client-side
- Ensures that client has received a message

**TIMELY**
- Not supported by Joram transport layer
ISIS-SOL mock-up Module Oriented Architecture

OSGi concepts and used technologies – Commanding Daemon

- OSGi: Java-based services framework, providing APIs and services
- Applications / components, organized as « bundles » (i.e. plug-ins)
- Equinox OSGi implementation
- Expected benefit: reduce complexity in most aspects of development
- ISIS-SOL mock-up Commanding Daemon = ISIS-SOL mock-up Daemon = OSGi application
- P2 Agent = OSGi application responsible for installations / updates
- Commanding Daemon composed of
  - Command control OSGi bundles
  - Common OSGi bundles
- Commanding Daemon uses
  - Declarative Service
  - Extension Point
- OSGi Container / JVM / OS
ISIS-SOL mock-up Module Oriented Architecture

OSGi bundles as major software structure elements – Commanding Daemon

Implementation Of Common Action Engine

Delegation of Telecommand management to LargeDataTransfer Daemon (large Telecommands)

Lower level services to upload telecommands

Log Management

Exposed MO Services initialization

Access to MAL Communication Layer

MOHelper OSGi Service

MAL OSGi Service

Log OSGi Service

Standard Telecommand Upload Service
Provides TC log report (in the ISIS-SOL mock-up)

Handles Telecommand Upload and Action Invocation service
Provides an abstraction to MAL layer and CCSDS MO CORE layer

Application Entry point To Commanding Component

Telecommand profile Encoding Service

Filter management for Telecommand Uploading

Command Control OSGi Bundles

Commanding Daemon OSGi Service

Commanding Encoding OSGi Service

Commanding Filtering Default OSGi Service

Commanding Filtering Large Data Transfer OSGi Service

Commanding Action OSGi Service

Command Control Common OSGi Bundles

Common Action OSGi Service

Common Action Engine OSGi Service

Common OSGi Bundles

MOHelper OSGi Service

MAL OSGi Service

Log OSGi Service

Exposed MO Services initialization

Access to MAL Communication Layer

Implementation Of Common Action Engine

Delegation of Telecommand management to LargeDataTransfer Daemon (large Telecommands)

Lower level services to upload telecommands

Log Management

The Power of Innovation

CS Communication & Systèmes – Charte 2010
ISIS-SOL mock-up Module Oriented Architecture

OSGi bundles as major software structure elements – Eclipse features

- ISIS-SOL mock-up uses Eclipse notion of features
- Feature = group of bundles, which defines the deployment strategy using P2 Repository mechanisms
- Separates logical software unit
- Guarantees that a group of bundles are coherent (software dependencies, pre-requisite specifications)
ISIS-SOL mock-up Module Oriented Architecture

Deployment characteristics and Eclipse Equinox P2 repository

- Deployment based on P2 Repository (Web server – HTTP Protocol)
  - All features are uploaded on this repository by an administrator

- After installation, every data exchanges go through the MAL (JORAM)

- High level of modularity
  - P2 repository allows easy upgrades on several machines, independently
  - Every component can be installed on any server
  - Scalability can be implemented easily (JORAM on a dedicated server for example)
Conclusions and first lessons learnt

- Although ISIS-SOL mock-up project is not finalized yet, some first lessons might be drawn from this mock-up.

- Integration of CCSDS MO standard and PUS standard: works pretty well!

- Our understanding and implementation of CCSDS MO Services corresponds precisely to how we imagined it worked!

- ISIS-SOL mock-up Architecture (owing to SOA) enforces inter-operability and saves cost and time during software integration.

- Notion of bundles and use of OSGi technology: interesting to implement and to guarantee modularity of software components. Only negative point: PDE Build (Eclipse build tool) is not very efficient to use when building large projects.

- Starting from a service-oriented description of the main control center functionalities, we have designed an architecture fully compliant with initial requirements.

Any questions?
ISI S-SOL mock-up:
Key Features Induced by the Adoption of New Standards

Annexes

1 - Documents and other references
2 - MO services: focus on the Command Control subsystem
   (external & internal interfaces)
Annex
Documents and other references

CCSDS standards
- CCSDS 521.0 : Mission Operations – Message Abstraction Layer
- CCSDS 521.1 : Spacecraft Monitor and Control - Common Services
- CCSDS 522.0 : Spacecraft Monitor and Control - Core Services
- CCSDS 133.0B1 : Space Packet Protocol
- CCSDS 910.3-G-3 : Cross-support concept part I - Space Link Extension Services

ECSS standards
- ECSS-E-70-41A : Telemetry and Telecommand Packet Utilization
- ECSS-E-ST-70-31C : Monitoring and Control Data Definition

OSGi standards
- Web Site : www.osgi.org
- OSGi Service Platform Release 4
Annex - ISIS-SOL mock-up Service Oriented Architecture
MO services: focus on the Command Control subsystem (external interfaces)
Annex - ISIS-SOL mock-up Service Oriented Architecture

MO services: focus on the Command Control subsystem (internal interfaces)