JSC / OTF
CCSDS Mission Operations
Project Status

Steve Lucord
October 31, 2011
Introduction

- iPAS
  - Implemented Command and Telemetry using CCSDS Mission Operations

- MCC21
  - Currently implementing private services using the CCSDS Message Abstraction Layer
Mission Control Center Role for iPAS

- To apply internationally standardized protocols to iPAS command and telemetry streams
  - Utilize Consultative Committee for Space Data Systems (CCSDS) standards
    - Encode MMSEV telemetry and command CCSDS Space Packets with the CCSDS Spacecraft Monitoring and Control (SM&C) protocol
    - Utilize the CCSDS Asynchronous Messaging System (AMS) to interface SM&C with CCSDS Delay / Disruption Tolerant Networking (DTN)
    - Transfer DTN data bundles over the space-to-ground segment with the CCSDS Licklider Transmission Protocol (ltp)
    - Adapt JSC MSKView display and control software to interface with SM&C
Mission Control Center Role for iPAS

- The iPAS is an application of ground-breaking work being done on integrating the entire SM&C / AMS / DTN / LTP protocol stack

• To begin to work with missions where light-time delay becomes significant
Telemetry Stream

CCSDS Packets (UDP)

CFS Lander (MMSEV)

On-Board Vehicle Sim

Transit Vehicle (Pass-thru)

Communication Environment – Telemetry and Command Laptop

Packet

CCSDS Packet Fan-out Utility (raw UDP data)

Test Orchestration

CFS-to-SM&C Converter/Provider Software

CCSDS AMS/DTN/LTP Software

DSNet

KSC LaRC ESTL

Control Center Simulation – Operations Technology Facility

JPL (Downlink Emulation)

CCSDS AMS/DTN/LTP

CCSDS SM&C Parameter Consumer

MSK Display
MSKView Telemetry and Command Display
Transport Broker

/Consumer /consumer:TransportAdapter /Transport (MessageBus) /provider:TransportAdapter /Provider

- REGISTER
- REGISTER_ACK
- NOTIFY
- DEREGISTER
- DEREGISTER_ACK
- subscribe(topic)
- defineTopic(if necessary)
- notify(topic)
- publish(topic)
- removeTopic(if necessary)
- unsubscribe(topic)
- PUBLISH_REGISTER
- PUBLISH_REGISTER_ACK
- PUBLISH_DEREGISTER
- PUBLISH_DEREGISTER_ACK
AMS Transport Broker

• AMS Role and Subject are defined in the MIB - are not dynamically added

• Role
  - Ground Zone, Session Name, Session Type, and Domain

• Subject
  - Area, Service and Operation

• MAL Abstract Broker filters subscription on the consumer side before issuing notify

• MCC21 Transport Broker will use middleware to filter subscriptions
MCC 21

- Mission Control Center for the 21\textsuperscript{st} Century
- Data Centric architecture
  - TIM (Tagged Information Message)
- Services defined and implemented using Mission Operations Message Abstraction Layer
- OMG Data Distribution Service (DDS) is used as the message bus
TIM Views

Application Layer

```c
struct A {
    int x_coord;
    int y_coord;
    struct B {
        float velocity;
    }
    struct C {
        char* x_label;
        char* y_label;
    }
}
```

Common Services Layer

```c
// Time t
/A/x_coord 1 t
/A/y_coord 2 t
/A/B/velocity 3.14 t
/A/C/x_label Time t
/A/C/y_label Ft/Sec t
```

Message Bus and Storage

**SDE Header**

- Name=/A , releaseTime= t
- protectionStatus=“group==telcom”

**Body**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_coord1</td>
<td>1</td>
</tr>
<tr>
<td>y_coord2</td>
<td>2</td>
</tr>
<tr>
<td>/B/velocity</td>
<td>3.14</td>
</tr>
<tr>
<td>/C/x_label</td>
<td>Time</td>
</tr>
<tr>
<td>/C/y_label</td>
<td>Ft/Sec</td>
</tr>
</tbody>
</table>

Application Object

Secure Data Envelope

TIM
Service Orchestration

- Data Access serves as the mediator to the underlying services that implement the logic
- Potential for multiple-inheritance for to implement many consumer adapters or provider handlers
- Use plug-in modules to encapsulate service consumer for each service.
Conclusion

• CCSDS Mission Operations been used to successfully in the MIRA and iPAS Projects

• Mission Operations system engineering concepts facilitated rapid specification of the MCC21 services

• MAL Framework jumpstarted development on the MCC21 project
  - Demonstrations are held at the end of each Agile iteration