OTF CCSDS Mission Operations
Prototype
Parameter Service

Phase I
Exit Presentation

John Stevens

02/25/2009
Project Goals

• Demonstrate the use of Mission Operations standards to implement the Parameter Service

• Demonstrate interoperability between Houston MCC and a CCSDS Mission Operations compliant mission operations center

• Utilize Mission Operations Common Architecture
Project Goals

Utilize Common Architecture

- Service Provider (Center A)
- Service Consumer (Center B)
- SM&C Standards (blue boxes)
  - MAL provides
    - Standard Patterns of Interaction
    - Fundamental definitions
  - Common Services provides
    - Standard Model for Services to Extend
    - Infrastructure Services
  - Mission Operations
    - Core Monitoring and Control, Planning, …
- Messaging Technology (e.g. AMS, SOAP, …)
Project Benefits

• Demonstration
  - Prototype CCSDS Blue Book service specifications
  - Pathfinder for inter-center communications

• Use of CCSDS Standards
  - International agreement
  - Reduce Cost

• Investigation of new technologies
  - Service Oriented Architecture (SOA) concepts
Project Definition

- Demonstrate Mission Operations Protocols
  - Scope
    - Implement the Core Services Parameter Service
    - Implement the Common Services Common Model Service monitorStatus Operation
    - Implement the Message Abstraction Layer (MAL)
    - Implement Information Sharing Protocol (ISP) for the transport layer
    - Interface with a Common Services Directory Service
Project Definition

- Out of scope (Related Services)
  - Core Services Check Service
  - Common Services Common Model Service Operations
    - request, list Status
    - request, list, monitor, add, modify, delete, and delete all Definitions
  - Core Service Parameter Service
    - set/get Filtered
    - set/get Timeouts
  - Common Services Configuration Management Service
    - Dependencies
  - OTF MO Common Services Directory Services Interface
Parameter Service Design

- Parameter Service Provider and Consumer Integrated Products
  - ANSI C
  - Based on the MCC Information Sharing Protocol (ISP) Client Toolkit
  - ANSI C GSOAP-XML Directory Service API
- Parameter Service Broker Integrated Products
  - ANSI C
  - Based on the MCC Information Sharing Protocol (ISP) Server Toolkit
  - ANSI C GSOAP-XML Directory Service API
ISP’ is a revised version of the ISP protocol for SM&C compliant messages.
Parameter Service - Interfaces

1 SM&C ANSI C Implementation (MAL, Common Services, and Core Services).
Parameter Service - Interfaces

2 SM&C ANSI C GSOAP-XML Implementation (MAL, Common Services).
3 SM&C Java JAXB-XML Implementation (MAL, Common Services).
Parameter Service - Structures

- NULL
- Element
  - Attribute
  - Enumerator
  - Container
- Enumerators
  - Parameter Service
    - Validity
  - MAL
    - InteractionType
    - QoSLevel
    - SessionType
    - UpdateType

Attributes
- MAL
  - Boolean
  - Blob
  - Double
  - Duration
  - FineTime
  - Float
  - Identifier
  - Integer
  - Long
  - Octet
  - Short
  - String
  - Time
  - URI

Containers
- MAL
  - Domain
  - EntityKeyList
  - EntityRequest
  - InteractionType
  - IdentifierList
  - MessageHeader
  - StandardError
  - Subscription
  - SubscriptionList
  - SubscriptionUpdate
  - SubscriptionUpdateList
  - Update
  - UpdateList
- Common Services (Common Model)
  - CompleteStatusUpdate
  - Status
  - StatusIdentifier
  - StatusUpdate
- Parameter Service
  - ParameterStatus
ANSI C Parameter Service

- ANSI C API Layer
  - Monitor Status operations support the Consumer-Broker-Provider interaction
    - SmcInitialize
    - CP_registerMonitorStatus
    - CP_deregisterMonitorStatus
    - CP_publishMonitorStatus
    - CP_notifyMonitorStatus
- MAL, Common Services, and Parameter Service Layers
  - Constructors, Destructors, Accessors
- Transport/Encoding Mappings
  - MCC ISP transport modified to support MAL PUBLISH SUBSCRIBE Pattern of Interaction
  - Binary Encoding of MO Structures

MAL PUBLISH SUBSCRIBE Message Pattern Sequence
Results

- Four (4) RIDS against MAL Book
- Two (2) RIDS against Common Services Book
- Two (2) RIDS against Core Services Book
- Three (3) generalized RIDS across the MAL, Common Services, and Core Services Books
Lessons Learned

• Significant amount of effort to modify legacy applications for SOA

• Message sizes significantly larger for Mission Operations than for legacy telemetry distribution
Conclusions

• Findings
  - To verify the CCSDS MO Standards meets manned spaceflight telemetry requirements, at a minimum the Parameter Service must be evaluated with the Check Service
  - Large amount of effort to modify legacy applications for SOA
  - Message sizes are significantly larger for Mission Operations
Contacts

• Management
  - Lindolfo Martinez (281) 483-4346 / 2099
  - lindolfo.martinez-1@nasa.gov

• Technical Lead (MO Prototype)
  - Walter Reynolds (281) 483-6723 / 2099
  - walter.f.reynolds@nasa.gov

• Responsible Engineer (Parameter Service)
  - John Stevens (281) 853-2343 / 595 Gemini
  - john.e.stevens@nasa.gov

• Project Sponsors
  - Eric Wolfer (281) 483-6709 / 2014A
  - eric.j.wolfer@nasa.gov
Thank You

• Questions
CCSDS Mission Operations
Parameter Service Demonstration

Screen Shots
1. Parameter Service Provider Log File
2. Parameter Service Broker Log File
3. Parameter Service Subscriber Log File

```plaintext
>>Decoded NOTIFY Message Header:
(msgHeader = {dataType = x; endianism = 03; URIfrom = "lance", authenticationId = "(9 bytes @ addr 9665b10) [anonymous]", URItos = "THERE", timestamp = 054/13:53:58.149; QoSlevel = 2, priority = 1, domain = "otf", "parmvec", "sm"), networkZone = "", session = 2, sessionId = "ISP_InControl SIM", interactionType = 6, interactionStage = 4, transactionId = "iscnull(6)"), area = "1", service = "1", operation = "8", version = 1, isError = 0}

>>Decoded NOTIFY Message Body: subscriptionUpdateList = subscriptionId = 7
timestamp = 054/13:53:58.149; sourceURI = lance; updateType = 2;
idList = {"SIN0000"}
entityId = (null); definitionId = (null); occurrenceId = (null); statusId = (null);
generationMode = 1; validityState = 2; rawValue = 64.944805; convertedValue = 64.944805
```

```plaintext
>>Decoded NOTIFY Message: subscriptionId = 7
timestamp = 054/13:53:58.149; (Body: subscriptionUpdateList)
```
4. Parameter Service Test Display