An Auto-Configuration System for the GMSEC Architecture and API

SMC-IT 2006

Joseph Moholt (ICS)
Arturo Mayorga (GSFC)
July 20, 2006
<table>
<thead>
<tr>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The Goddard Mission Services Evolution Center (GMSEC)</td>
</tr>
<tr>
<td>• Automated Configuration Concept</td>
</tr>
<tr>
<td>• Implementation Approach</td>
</tr>
<tr>
<td>• Key Components and Benefits</td>
</tr>
</tbody>
</table>
GMSEC

- The Goddard Mission Services Evolution Center (GMSEC) was established in 2001 as an evolutionary approach to ground system architectures

- Objectives
  - Simplify integration and development using a Plug and Play approach
  - Facilitate technology infusion over time
  - Support evolving operational concepts
  - Avoid vendor lock-in

- Key Concepts
  - Standardize interfaces – not components
  - Provide a middleware infrastructure
  - Allow customers to select components that meet their needs
GMSEC Ground System Architecture

Traditional Design
Socket Connections

GMSEC Design
Middleware Connections

Middleware simplifies integration by having components interface to a bus and not to each other
GMSEC API

- Application Program Interface (API)
  - Provides connection between application components and architecture middleware
  - Typically implemented as a function library
  - One API implementation per language

- Encapsulate messaging transport
  - Single API for multiple middlewares
  - Flexibility for middleware additions
  - Platform and operating system independence

- Abstract messaging structures
  - Simple message model
  - Facilitates data packaging
Extending GMSEC Concepts through Automated Configuration

- Currently, configuration of a GMSEC-based system requires separate configuration for every component
- An automated, centralized configuration will provide for easier management of architecture components
  - Define a standard for managing the configured content
  - Facilitate reconfiguration of components
  - Allow faster configuration for new missions
Drivers

- The Automated Configuration System must
  - Be platform and language independent
  - Provide a standard data representation format
  - Provide an efficient data representation format
  - Support client/server and server/server communications
  - Provide broadcast capability
GMSEC Ground System Configurations

Current GMSEC Configuration

Automated GMSEC Configuration

Centralized configuration agent simplifies integration by having components interface to a single configuration point.
Impact to Communication Layers

Auto-configuration approach affects the top four layers of the Open Systems Interconnect (OSI) model

- Application Layer
  - Provides service to end user
  - Impacts the GMSEC API

- Presentation Layer
  - Provides data representation
  - Impacts data packing definitions

- Session and Transport Layers
  - Provides connection/data format
  - Impacts protocol selection
Implementation Approach

- Create a Configuration Management Agent (CMA)
  - Implemented as a GMSEC component
  - Manages configuration files

- Enhance GMSEC API to automatically interact with CMA
  - Provides a transparent way to retrieve configurations from CMA

- Create a lightweight middleware
  - Limited to required features
  - Facilitates communication between API and CMA

- Include both TCP and UDP to meet communication requirements
  - TCP for clients
  - UDP for inter-server (broadcast)
Auto-Configuration System Diagram

- CMA
- MBServer
- UDP Broadcast
- GMSEC Message Bus

Temporary TCP connection

New GMSEC component

GMSEC component

GMSEC component

COTS Mission GMSEC Bus
Key System Components

- Configuration Management Agent
  - Holds component configurations
    - Middleware options
    - Message formats
    - Component specific options
  - Responds to configuration requests
- Enhanced GMSEC API
  - Retrieves component configuration from CMA
  - Provides the interface to component developer
- GMSEC Message Bus as the middleware solution
  - Self-configuring, hub and spoke middleware
  - Supports inter-server communication
  - Bridges CMA and GMSEC API enhancement

Although designed as an integrated system, components can also be used independently
Benefits of the Auto Configuration Approach

- Significant reduction in integration time
- Components added/upgraded/migrated without impacting existing system
- Standard configuration approach provides reuse of configuration specs between different missions
- Vendors can build on the configuration definition standards to more easily integrate their products into the NASA ground system domain
Summary

- The GMSEC architecture provides a scalable, extensible ground and flight system approach for future NASA missions, enabling easy integration of components to meet customer requirements
  - Open source download at http://opensource.gsfc.nasa.gov

- The Auto Configuration System expands on GMSEC’s service oriented architecture by providing further integration and configuration capabilities
  - Currently being deployed to component developers
# Acronym List

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>CMA</td>
<td>Configuration Management Agent</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off The Shelf</td>
</tr>
<tr>
<td>GMSEC</td>
<td>Goddard Mission Services Evolution Center</td>
</tr>
<tr>
<td>GSFC</td>
<td>Goddard Space Flight Center</td>
</tr>
<tr>
<td>ICS</td>
<td>Interface &amp; Control Systems</td>
</tr>
<tr>
<td>MB</td>
<td>Message Bus</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnect</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
</tbody>
</table>
Questions?

jmoholt@interfacecontrol.com