The Benefits of a Service Oriented Architecture for Flight Dynamics Systems

Brian Robinson
a.i. solutions, Inc
3/12/2010
Agenda

• Introduction to the Problem and Solution

• Example SOA Systems Implemented
  – Goddard Trajectory Determination Software
  – Risk Reduction/FreeFlyer Demonstration
  – LandSat Data Continuity Mission Flight Dynamics System

• Summary
Current Spacecraft Ground System Problems

• Spacecraft operational life is usually extended well beyond the primary mission
  – In the operations environment there is a heavy opposition to change
    • If it isn’t broken don’t fix it
  – Results in aging or unsupported software
    • Required upgrades introduce risk

• Ground Systems are built on many different components
  – Each component must speak to the others
  – Requires significant effort to develop and maintain interface control documents as well as each components software interface
Solution

• Service Oriented Architecture
  – Specifically utilizing the plug-and-play features
    • Interfaces can be made “generic” – no need for “component” specific interfaces
  – Provides a robust capability to update system capabilities as mission requirements change or evolve
  – Eliminates risk associated with the use of non-configuration controlled off-line tools
  – Addresses the issue of maintaining unsupported software
  – Provides effective two-way communications between system processes
  – Greatly decreases the development and maintenance burden of ICDs.
Examples

• Interface Goddard Trajectory Determination System (GTDS) with the Goddard Mission Services Evolution Center (GMSEC) Message Bus as a demonstration of the ability to rejuvenate legacy applications

• Extend capabilities of the COTS FreeFlyer to support GMSEC and the Oracle Service Bus (OSB) in a demonstration of ground system design in the dark

• Design the LandSat Data Continuity Mission (LDCM) Flight Dynamics System to support SOA features
Goddard Trajectory Determination Software

- GTDS is a critical component to the Flight Dynamics Facility (FDF) at NASA Goddard
- The software is 30+ years old and is expected to be used for the foreseeable future
- The FDF is re-engineering to a SOA system utilizing GMSEC
- Current automation processes for GTDS and other legacy applications will not be effective in the new environment
Two modules were created as part of a demonstration to integrate legacy applications with the GMSEC system:

- A generic module which will be used by any and all FD applications:
  - Supplies the connection to the GMSEC Message Bus
  - Standardized interface for any flight dynamics software
- An application-specific module ties the GTDS software to the generic module to complete the interface:
  - Translates to native input/output format
  - Starts the GTDS application
  - Supplies information to be sent back to the GMSEC bus
Lessons Learned

• Legacy applications can be integrated with SOA systems to meet evolving ground system requirements

• It is a large effort to interface complex and legacy software like GTDS with SOA systems
  – There are hundreds of keyword cards and thousands of options that must be accounted for in order to effectively integrate the software.
  – Required information in messages may create an application specific message
**FreeFlyer Demonstration**

- FreeFlyer provides the user the ability to extend its capabilities via custom DLLs.
- Extensions to the FreeFlyer base code (DLLs) were created to support interfaces with both GMSEC and Oracle Message Buses.
- The GMSEC extension was used in a demonstration that showed that the different ground system components could be designed in a near vacuum and integrated using the message bus.
- The Oracle extension was used in a risk reduction effort to show that the different subsystems (FDS, TT&C, and Mission Planning) could be integrated with the Oracle Bus.
Lessons Learned

• The capabilities of FreeFlyer can easily be extended to support SOA systems such as the GMSEC and Oracle Services Message Busses
• It is possible to develop a ground system component in a near vacuum, but some insight into the other systems is still necessary
• There should be a significant reduction in development of interface controls and documentation
The LDCM Mission Operations Center (MOC) is being developed around a SOA-like system.
  - All information (Heartbeats, Statuses, and Data) is transferred via files using a common bus

The FDS system supports the MOC SOA system
  - Receives and reacts to incoming files
  - Creates files that initiate activities in other sub-systems

The FDS is a SOA within a SOA
• A central system, the Service Manager, manages receipt of new files
• Based on the type of file the system executes a sub-process and continues to wait for new files
• The Service Manager also reports the status of the files and the sub-processes
  – Log files and heartbeats are created
  – Event messages are created at each level
Lessons Learned

• While the MOC is not employing a SOA system in a strict sense, there is advantage in maintaining a SOA model

• The plug-and-play functionality built into the FDS system provide superior flexibility to standard FDS systems
  – Flight Ops Team can add or remove actions as needed
  – New software and updates to existing software can be tested without disrupting operations
Summary

- SOA systems can be integrated with legacy software to enhance their usefulness
- SOA systems can reduce cost associated with developing a ground system
- SOA systems enable software testing and integration to occur in a much less invasive manner