Useful Sensor Web Capabilities to Enable Progressive Mission Autonomy

Dan Mandl

Agenda

- Statement of challenges
- Capabilities to meet challenges
- Experiments run with real satellites to validate new capabilities
  - EO-1
  - CHIPS
  - ST-5
- Conclusion
Challenges for Future Missions

- Manage increased mission complexity at lower cost
- Create flexible missions with interoperable components
- Increase mission safety by embedding intelligence to manage security and hazard avoidance

Key Capabilities To Meet Challenges

- Transition from centralized mission control to distributed control
- Maximize interoperability by abstracting as much mission functionality as possible
- Develop and use self-managing software components (autonomic computing)
  1. Components have self-awareness
  2. Self-optimization
  3. Self-healing
  4. Self-protection
  5. Negotiates (peer-to-peer) for resources
  6. Functions in a heterogeneous world and with open standards
  7. Anticipates needed resources and hides details needed to obtain resources
Series of Experiments Conducted

- Used following missions to conduct experiments to facilitate these capabilities:
  - Earth Observing 1 (EO-1)
  - Cosmic Hot Interstellar Plasma Spectrometer (CHIPS)
  - Space Technology 5 (ST-5)
  - Experiment with Service Oriented Architectures (SOA)

EO-1 Satellite
Launched November 21, 2000

NASA New Millennium Program space technology validation mission
- Hyperion – hyperspectral instrument
- Advance Land Imager (ALI) – multispectral instrument
- 10 other space technologies validated
- 2 Mongoose onboard computers with 256 Mbytes each

Presently in extended mission and being used for additional experiments with hyperspectral imagery and sensor web experiments
Cosmic Hot Interstellar Plasma Spectrometer (CHIPS): A Flying Networked Computer Testbed for Advanced Mission Concepts

- PowerPC
- Onboard IP Stack
- 128 Mbytes of Memory
- Perfect for Experiments (E.g. Secure IP to S/C)

**OPEN FOR BUSINESS 9/1/05 **

---

**ST-5 Overview**

ST-5 is a three satellite (micro-sat) constellation
NASA New Millennium Program technology validation mission
Help scientists understand the Earth’s magnetosphere and its effect on space weather
Uses the GMSEC architecture to enable cost-effective model-based operations to run the ST-5 constellations lights out
Launched March 22, 2006
Successfully completed mission June 22, 2006

Not much bigger than a large birthday cake or a small TV.
E.g. - EO-1 Targets National Priority Wildfires

Identify NIFC-tracked Wildfire Incidents
Large Incidents - August 21, 2003

Aqua or Terra MODIS data

SGM
Correlate latest fire location information with MODIS Imagery
Roberts Fire

Active Fires Detection Map

UMD Natural Hazards Investigation Team

USFS Burned Area Fire Danger Response Team

Fire location confirmed and selected for imaging

SGM adds target to EO-1 ground & on-orbit planning & scheduling systems and tasks EO-1

ERD Data Center
L1 Data

Vision to Enable Sensor Webs with “Hot Spots”
Sensor Web Experiments, Event-driven Observations, Onboard Autonomy

Software-driven Antenna Synthesis Patterns - Software Tunes Reception to Targeted Satellite
Plug-and-Play Flight Software
Underlying "Plug and Play" Message Bus Architecture -- Goddard Mission Services Evolution Center (GMSEC)

GMSEC architecture provides a scalable and extensible ground and flight system approach:

- Standardized messages formats
- Plug-and-play components
- Publish/Subscribe protocol
- Platform transparency

More info at: http://gmsec.gsfc.nasa.gov

GMSEC Component Catalog

GMSEC approach gives users choices for the components in their system. The TRMM mission has selected key components from the GMSEC catalog:

- Front End Processors + Simulators
- Scripting Control
- Telemetry + Command Systems
- GMSEC APIs/Ops Systems

COTS Middleware: SCL, Tibco Rendezvous, Tibco SmartSockets, Elvin

- Automation
- Archive + Assessment
- Flight Dynamics
- Planning + Scheduling
ST-5 Lights-Out Autonomy

ST-5 mission demonstrated parts of (1), (2), (5), (6) and (7) (from slide 3)
- Lights-out operations with model-based software
- Predict problems before they happen and fix early
- Models update themselves automatically
- Modeling system is built on top of "plug and play" architecture to enable easy extensibility
- Act as stepping stone for this type of capability for future missions
**ST-5 ROME Framework**

- Real-time Object Modeling Executive (ROME)
  - Supports multiple models and multiple spacecraft
  - Leverages common engineering modeling environments
  - Models from various sources are easily integrated
  - Fully supports GMSEC bus
  - Models initialized and maintained from telemetry
  - Model control via configuration file or bus directive
  - Results available to GMSEC subscribers
  - Easily configured via XML
  - Highly scalable

**ST5 Specific Configuration**

- ROME based implementation
- Dynamic characterization of sub-systems phenomenology
- Used by mission to manage constrained resources
- Models of Subsystems
  - Solid State Recorder (Contractor Dev.)
  - RF (Student Dev.)
  - Power (Project Dev.)
SimulinkST5 GMSEC Highlights

- Simulink is a visual interface to MatLab to allow users to simulate systems that can be represented with mathematical equations.
- Features of Simulink as used on ST5 are as follows:
  - Standardized messaging interoperability
  - GMSEC Compliance
  - Directives
    - Advanced Mission Planning System (AMPS)
    - Advanced Spacecraft Integration and System Test (ASIST) system
  - Mnemonic Value Messages
    - ASIST
    - Integrated Test and Operations System (ITOS) capability
  - Heartbeat messages
  - Log messages
  - Product Messages
- Predictive Model-Based Operations
- Subsystem models to anticipate platform conditions in a constellation environment.
- Support Short and Long Term Mission Planning
- Interact with AMPS and ASIST for control directives, telemetry, and profile events.
- Constellation Operations Support

Core Flight System (CFS) and Extension for GMSEC for Flight SW

CFS provides a framework that simplifies the development and integration of applications

- Layered Architecture – software of a layer can be changed without affecting the software of other layers.
- Components communicate over a standard message-oriented software bus, therefore, eliminating the need to know the details of the lower layers of inter-networking.
- Software components can be developed and reused from mission to mission.
- Developed by Flight SW Branch at GSFC

More info at: http://gmsec.gsfc.nasa.gov
Onboard Integrated Message Bus Demonstration (December 2005)

Ground System Testbed

- ASIST Primary
- ASIST Secondary
- GMSEC Bus

cFE on CHIPS

- Command Ingest
- Telemetry Output
- Livingstone Adaptor
- Model
- Script
- Result

DC = Data Center
ASIST = Advanced Spacecraft Integration and System Test

Moving Models Onboard CHIPS Satellite Under cFS to Demonstrate Mobile Agents

- Mobile agent – autonomous software module that can easily be moved around a network
- Models transformed into mobile agents
  - Worked with Solid State Recorder agent (model) first
- Adapter built to make compatible with both GMSEC and Core Flight Executive (cFE)
- Demonstrated capability to move software running on GMSEC onboard to run under cFE
- Demonstrates beginning step to transform missions from central control to distributed control via self-managing software
One of Three Experiments Conducted by UMBC Undergraduate Class 12-14-05

Experiment with UMBC Undergraduate Class 12-14-05

Picture of Experiment Day

Sensor network class, Dr. Younis, Vuong Ly and Dan Mandl

Sensor mote layout & atrium where experiment conducted (inset)

Mini-rover in action

Baltimore Sun reporter

Mini-rover autonomously finding broken sensor node (part of Emergency Response UMBC project team)
Closer Look at our Mini-Rovers & Simulated Mars Landscape at GSFC

Began to Implement Experiments with Standards - Vision for Integrated Sensor Web Environment

- GSFC Mission Systems Evolution Center (GMSEC)
- Core Flight Executive (cFE)
- Core Flight System (CFS)
- SensorML
Sensor Modeling Language (SensorML)

- Standard models and Extensible Markup Language (XML) schema
  - Describes sensor systems to provide information needed to discover and locate sensor and sensor observations
  - Process low-level observations
  - Defines interfaces
  - Lists taskable properties
- Can apply to any sensor whether in-situ or remote
- Facilitates “plug and play” and interoperability between sensors
  - Especially useful for heterogeneous sets of sensors and rapid integration of new sensors

More info at— http://vast.nsstc.uah.edu/SensorML/

OGC EO-1 Experiment

- A proposal was submitted and accepted by the Open Geospatial Consortium (OGC) to use EO-1 as part of a testbed effort beginning June 2006 and lasting until December 2006.
  - Testbed effort called OGC Web Services (OWS) – phase 4 has may objectives, one of which is Sensor Web Enablement (SWE) for sensors via a standard which is similar to a Service Oriented Architecture (SOA)
  - More info at:
    http://www.opengeospatial.org/initiatives/?iid=199
  - Sponsored by many organizations including NASA, NGA, GeoConnections – Canada, National Technology Alliance, GSA, ORNL, LMCO, BAE, Ordinance Survey – UK, NATO C3, and TeleAtlas
OGC EO-1 Experiment

- Figure on next slide depicts the portion of the demonstration in which EO-1 will participate: generic capability to discover and task EO-1 on the Internet via the following services:
  - Sensor Planning Service (SPS) – a standard Web service interface for requesting, filtering and retrieving sensor observations
  - Sensor Alert Service (SAS) – a standard interface for asynchronous notification of messages or alerts from sensors or sensor services
  - Sensor Registration Service (SRS) – a standard Web service to store sensor characteristics for later user retrieval
- Geospatial Interoperability Office at GSFC (M Bambacus, Nadine Alameh) major sponsor of OGC activities (and in particular OWS-4) and are monitoring this activity

EO-1 Participation in OGC Web Services – Phase 4 Testbed; June 2006 – December 2006

Open Geospatial Consortium (OGC) Web Services
- Phase 4 Testbed (June – December 2006)
- EO-1 portion of demonstration

- Search for satellites based on capabilities
- Register sensor in catalog to make it searchable on Internet
- Command / send tasking requests via Sensor Planning Services (SPS) protocol
- Request alert service from satellite via Sensor Alert Service (SAS)
- Wrap EO-1 satellite in SensorML and publish its capabilities
- Enable generic command / tasking request via SPS
- Enable generic alert services via SAS

- Store capabilities
- Process user query and return the result
Conclusion

- Building capabilities to enable progressive mission autonomy via the use of three satellites and a series of increasingly more capable experiments
- Focusing on validating distributed mission control and maximizing interoperability
  - Enable changes to mission post launch
  - Combine existing missions into temporary "virtual constellations" to enable new missions

Conclusion

- Will add additional real experiments to continue to build the toolbox
  - Two recent awards for AIST ESTO call for proposal will be used
    - An *Inter-operable Sensor Architecture to Facilitate Sensor Webs in Pursuit of GEOSS – Related to OGC effort*
      - Key topic - Interoperability
      - PI: Dan Mandl - 3 year effort
    - *Using Intelligent Agents to Form a Sensor Web for Autonomous Mission Operations*
      - Key topic distributed mission control
      - PI: Ken Witt/ISR Co-I Dan Mandl/GSFC - 3 year effort
Conclusion

- Other related awards from AIST ESTO call in March 2006 will allow possible further synergy
  - E.g. - *Increasing the Technology Readiness of SensorML for Sensor Webs*
    » Key topic SensorML for sensor interoperability
    » PI- Michael Botts/Univ. of Alabama – 3 year effort