

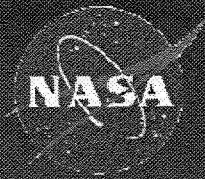
Manhattan Beach, California March 28-30, 2006

Tool Development for Distributed System Architectures

Thomas Grubb
NASA/Goddard Space Flight Center
Thomas.G.Grubb@nasa.gov
GMSEC@nasa.gov

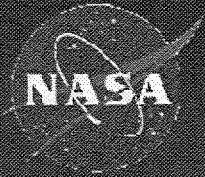


Tool Development for Distributed System Architectures



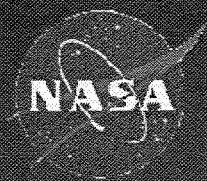
Purpose

- To show how a message bus architecture facilitates growth and change, and enables opportunities for rapid application development
- To show tools developed at NASA Goddard Space Flight Center for the Goddard Mission Services Evolution Center (GMSEC) architecture



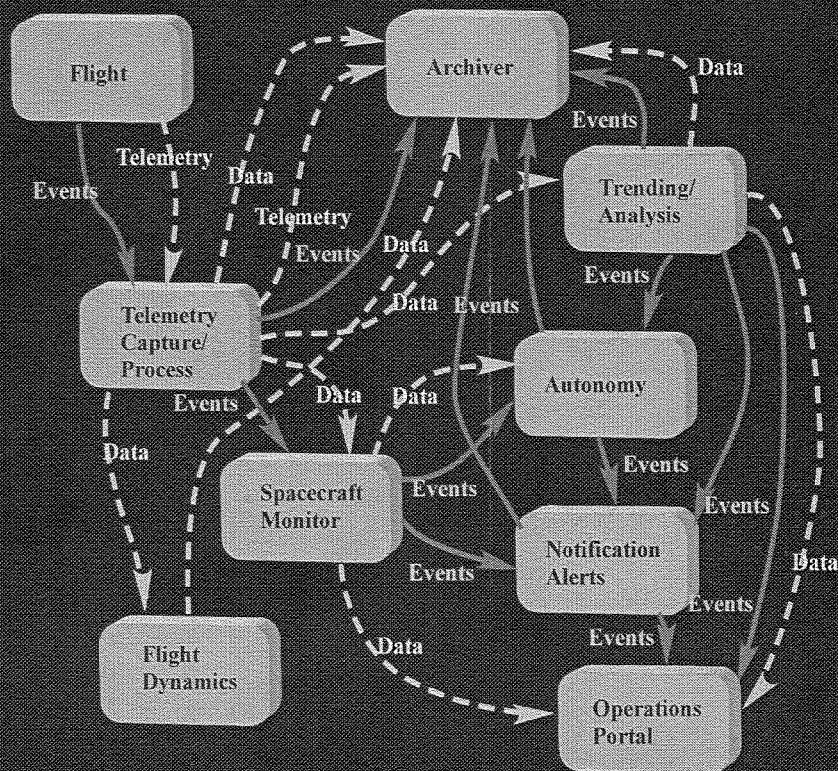
What is GMSEC

- **GMSEC** (Goddard Mission Services Evolution Center)
 - Developed to improve how NASA would develop and maintain ground data systems for dozens of missions, with a constant stream of missions always in the development phase.
- **Four Key System Concepts**
 1. Standardize interfaces (not components). Loosely coupled
 2. Utilize a messaging middleware to develop a framework (publish/subscribe)
 3. Provide the user with choices for major functional components
 4. Own the reference architecture and interface standards, allow vendors and development organizations to own and advance their functional areas

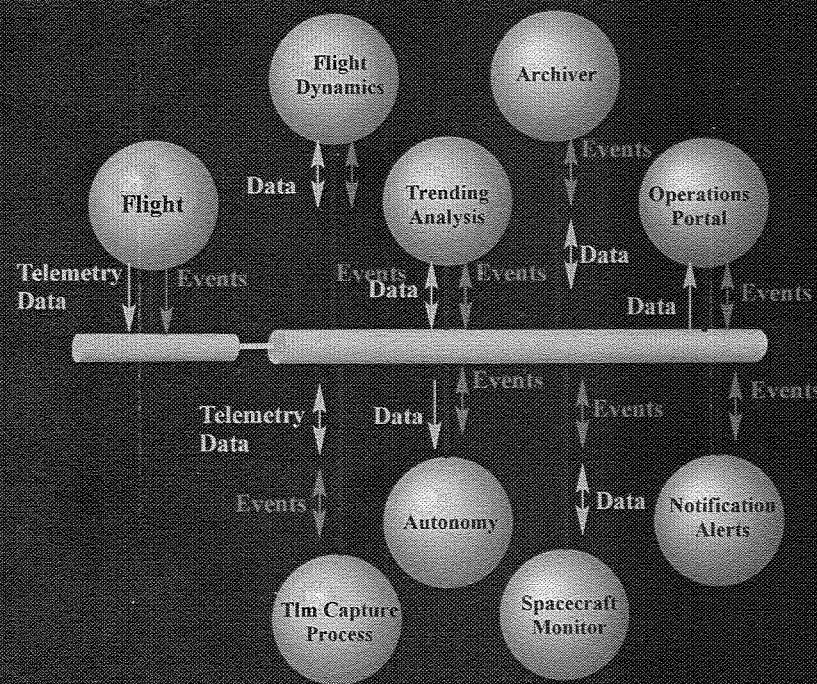


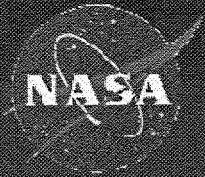
Interface Standards and Middleware Simplifies Architecture

Traditional Design Socket Connections



GMSEC Design Middleware Connections





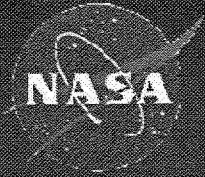
What GMSEC Framework Provides Developers

■ Standard API

- Multiple languages (C, C++, Java, Perl, Python, etc)
- Multiple middlewares (SmartSockets, Rendezvous, MagicBus, etc)
- Multiple platforms (Windows, Linux, Solaris, Mac OS)

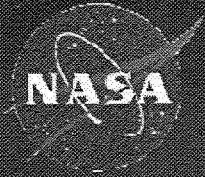
■ Standardized Messages

- Reduced programming integration (e.g., don't have to worry about the applications, just the messages. In some ways, Similar to blackboards and/or goal based architectures)
- Reduced system integration (applications can be added/deleted/swapped on the fly)



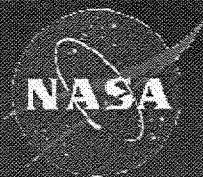
What this means for developers

- Tools can be developed with minimal knowledge or impact, on other applications
 - Enables opportunities for developers to quickly, rapidly, and cheaply develop applications for targeted problems and/or system-wide areas.
- Tools can be added, deleted, and updated easily while a system is running
- Generalized tools can easily be developed, with minimal migration effort from mission to mission
- High return on development investment by using the strengths of your team as regards to platform, language, and product knowledge to add value to systems



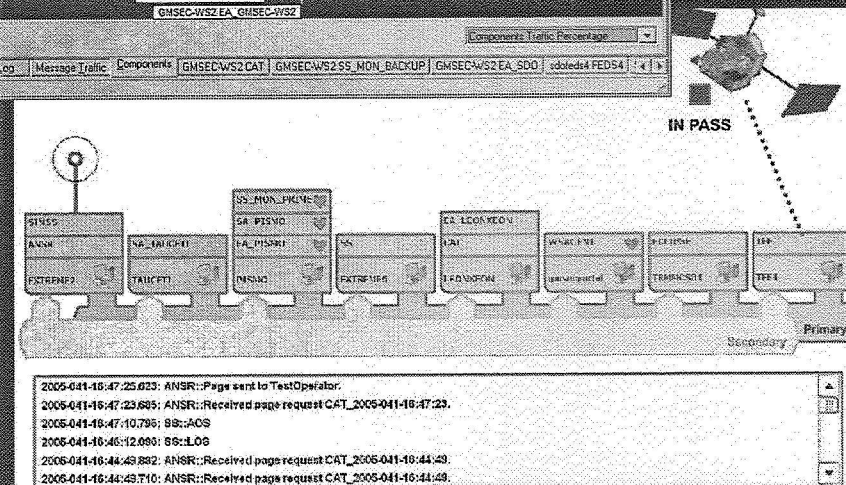
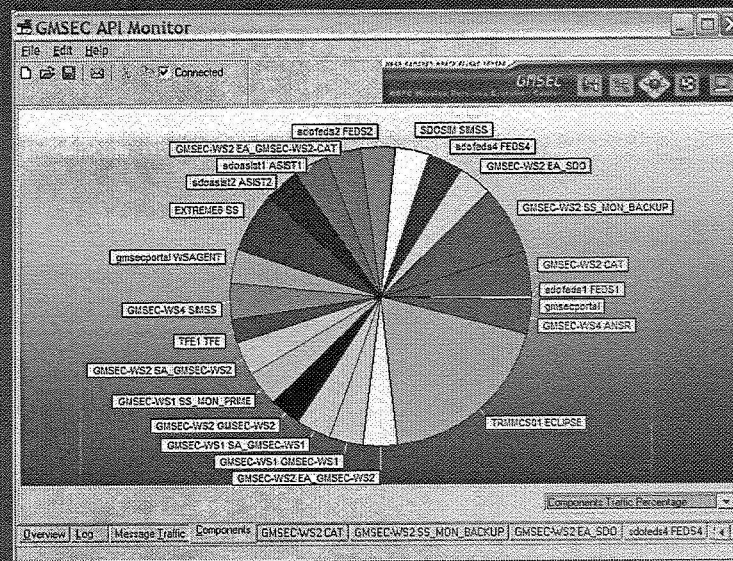
Types of Tools

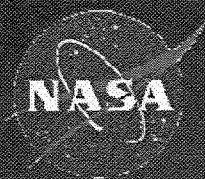
- Monitoring applications for displaying components on the bus, their status, and communication statistics.
- Debugging/Simulation applications for recording and then playing back any type of message traffic on the bus.
- Web Services for remotely accessing a GMSEC bus from ANY language, ANY platform, and ANY location.
- Automation applications for automatically scheduling activities or analyzing and reasoning about components and the system.
- Logging and Archiving applications for displaying, reporting, and saving telemetry, event/log, directive, command or any other types of messages from the bus.
- Commanding
- New Middlewares



Health Monitoring – GSMO, GEDAT, and API Monitor

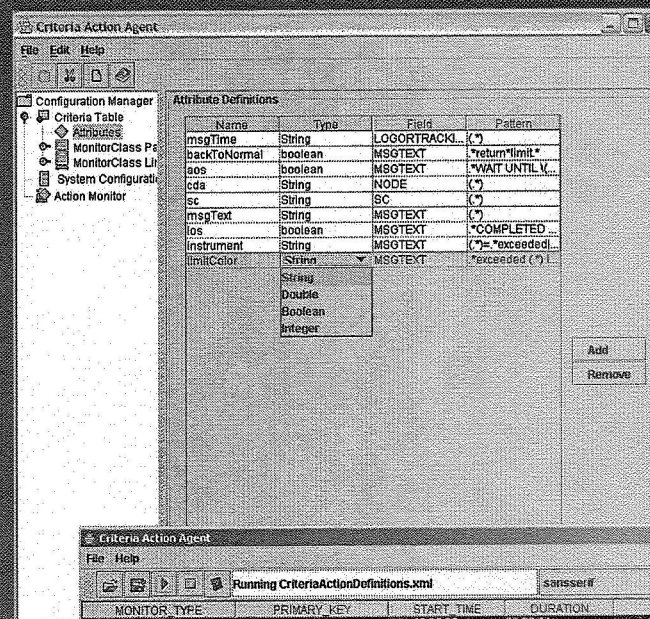
- System-wide real-time application health monitoring
 - Visual overview of machines and components on the GMSEC bus
 - Monitors published messages on the bus, including heartbeats
- Required no changes to other applications
- Extremely small development effort
 - API Monitor – Windows-based Health Monitor and graphs (< .1 FTE)
 - GSMO – Flash-based Overview Display (.1 FTE)
 - GEDAT – Java-based Overview Display .375 FTE)





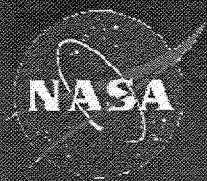
Analysis & Response – System Agents & Criteria Action Table (CAT)

- Rule based fault detection and response
 - Provides more automation and autonomy to mission operations
 - Monitors Heartbeat and System Agent messages to automate detection of anomalous conditions
 - Works with System Agents to detect and respond (failovers) to problems
- Required no changes to other applications
- Moderate development effort: System Agents (.1 FTE), CAT (1.5 FTE)



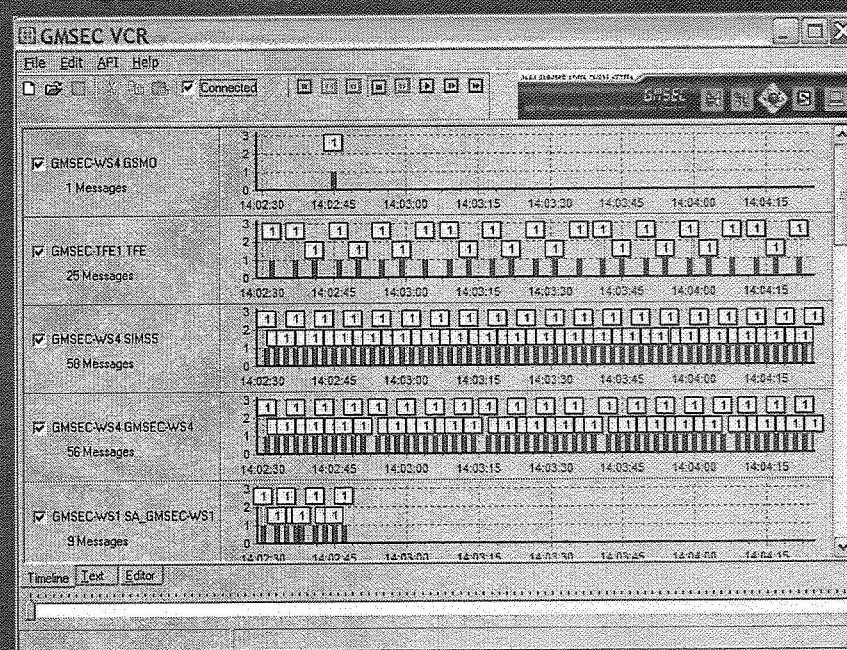
The screenshot shows the 'Criteria Action Agent' running table, displaying a list of monitors and their status.

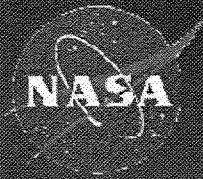
MONITOR TYPE	PRIMARY KEY	START TIME	DURATION	STATUS
HeartbeatMonitor	CAT	111-19:25:23	00:00:11	running
HeartbeatMonitor	EG	111-19:25:32	00:00:32	running
PosMonitor	SC12	111-19:25:43	00:00:21	action taken on 111-19:26:04, still running
LimitViolationMonitor	SC12JIPSM4 1	111-19:25:48	00:00:16	action taken on 111-19:25:59, still running
LimitViolationMonitor	SC12.GFAILTYPA	111-19:25:58	00:00:09	running
LimitViolationMonitor	SC12.QGPHRATE	111-19:26:03	00:00:02	running



Simulation, Presentation and Testing Tools – GMSEC VCR

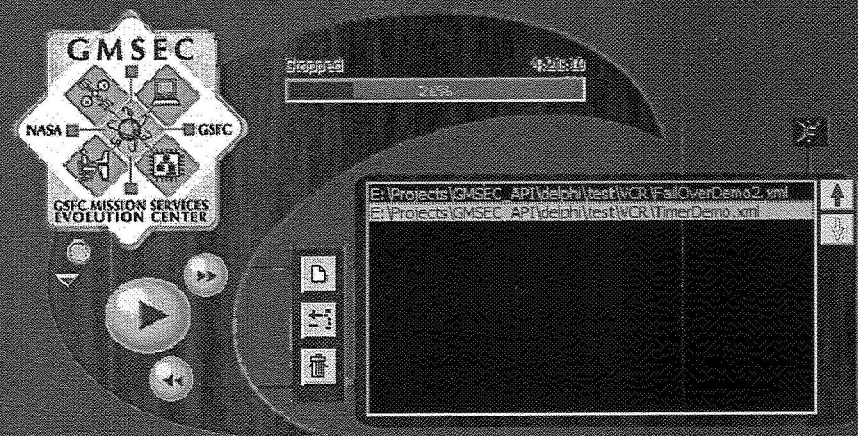
- Recording and playback of published messages
 - Extremely helpful in debugging and simulations
 - Similar to a multitrack tape recorder where each component is a separate track.
 - VCR Functions (Record, Play, Stop, Pause, Step Forward, Step Backward, etc) and Editing Functions (Add, Edit, Delete messages)
- Required no changes to other applications
- Extremely small development effort (< .1 FTE)

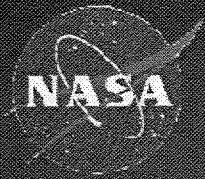




Simulation, Presentation and Testing Tools – GMSEC Symphony

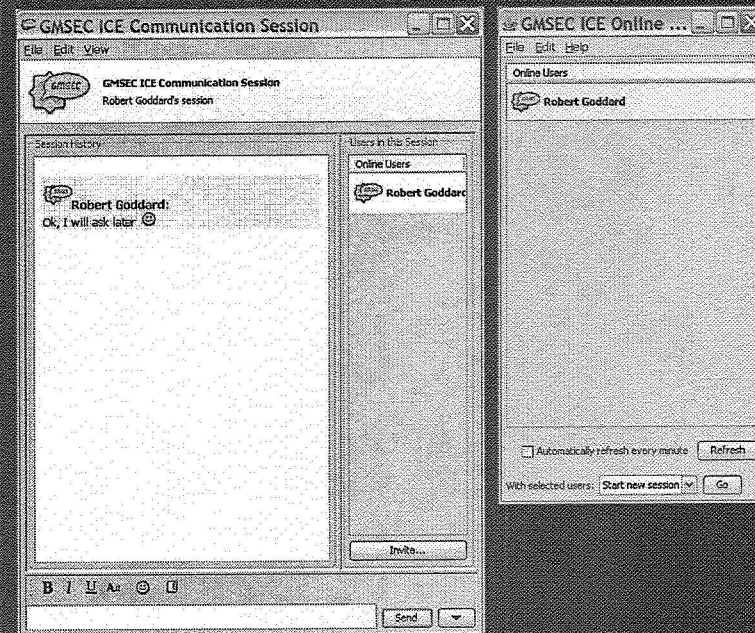
- Playback of lists of recorded sessions
 - Excellent for presentations
 - Uses a familiar paradigm for managing and playing back a play list of GMSEC VCR files
 - Manage a “play list” of VCR files: Add, delete, and reorder your playlists and save them for later
 - Play each VCR file in sequence, loop them, pause after playing each file, and jump to specific points in a file
- **Required no changes to other applications**
- **Extremely small development effort (1 week)**

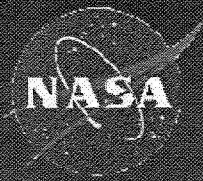




Simulation, Presentation and Testing Tools – GMSEC Integrated Communications Environment (ICE)

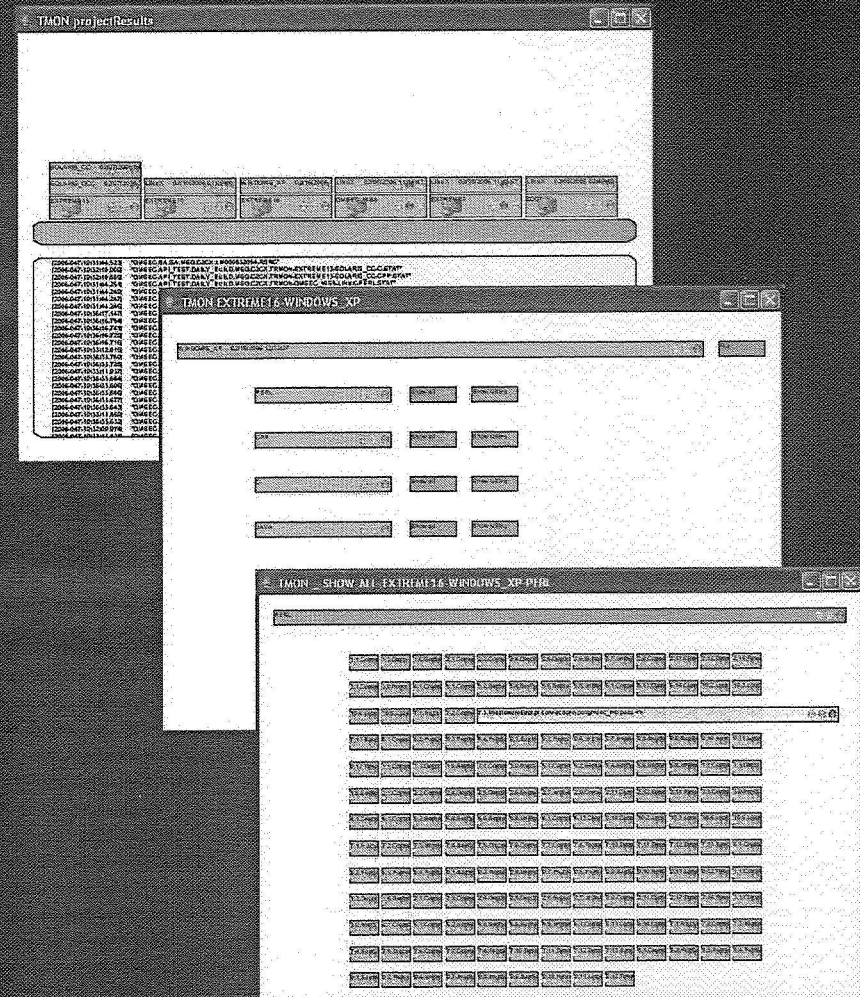
- Text conferencing solution using the GMSEC Bus
 - As secure as your network
 - Provides Full-featured text conferencing with graphical "emoticons", WYSIWYG text editor, file attachments, and user icons
 - Can be used as an application or in a browser and part of GMSEC Portal
- Required no changes to other applications
- Extremely small development effort (.1 FTE)

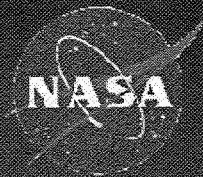




Simulation, Presentation and Testing Tools – GMSEC Testing Suite

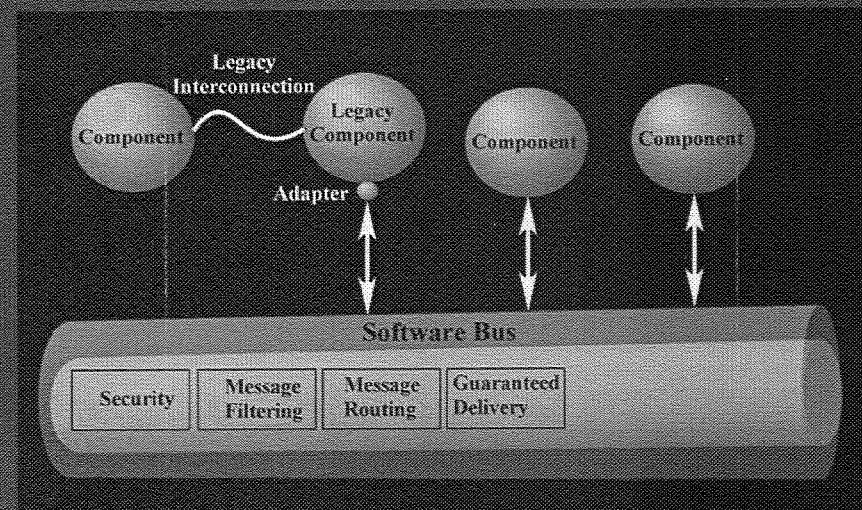
- Automated Testing Suite for GMSEC API run on a daily basis
 - Tests the latest development version of source code
 - Daily builds and tests, generate result reports
 - Total of 91 test cases per middleware wrapper
 - Currently Releasing/Supporting
 - 3 wrappers * 4 languages * 6 platforms
 - 6552 test permutations to execute for the entire suite
- Moderate development effort (.5 FTE)

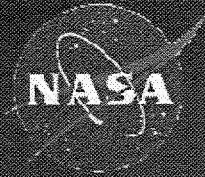




Middleware and Bridges – GMSEC Message Bus (MagicBus)

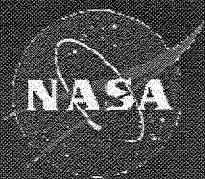
- Light weight middleware implementation
 - Supports all messaging capabilities of the GMSEC API.
 - Excellent for development period and small systems
 - Easy to use and configure
- **Required no changes to other applications**
- Small development effort (.375 FTE)





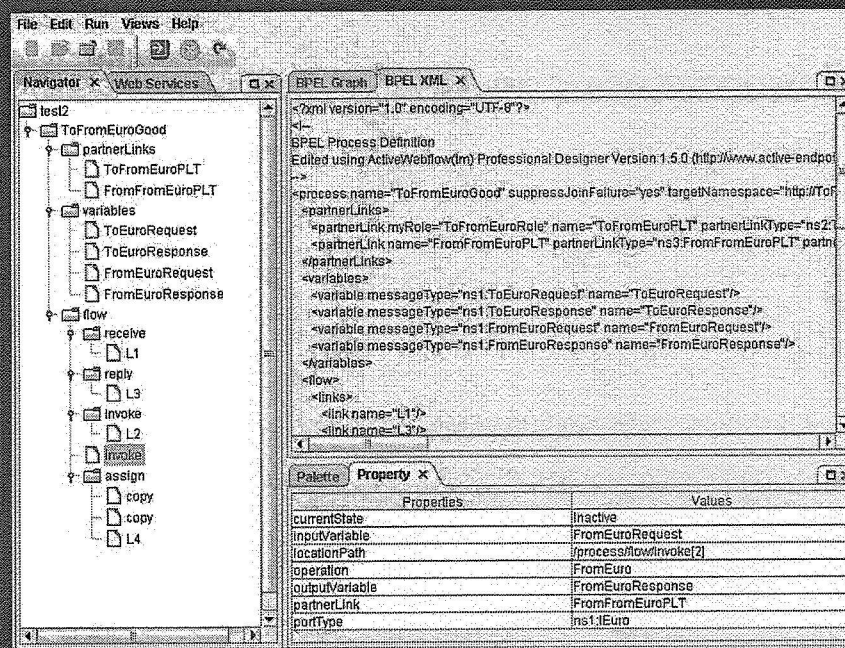
Middleware and Bridges – GMSEC Bridge

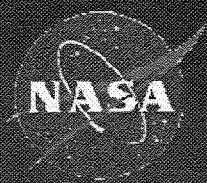
- Bridges two GMSEC buses to make one virtual bus
 - Transparently bridges messages between two (or more) GMSEC buses
 - Provides message filtering
- **Required no changes to other applications**
- Small development effort (.3 FTE)



Middleware and Bridges – GMSEC Web Services

- Exposes GMSEC Bus through web services
 - Enables integration of GMSEC system architectures with non-GMSEC systems
 - Enables GMSEC enabled applications for new platforms and languages
 - Can be integrated into web applications for the GMSEC portal with zero-configuration for clients
- Required no changes to other applications
- Extremely small development effort (.2 FTE)





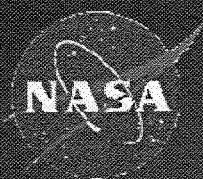
Logging and Archiving – GMSEC Reusable Events Analysis Toolkit (GREAT)

- Log Archiver and Reporting Tool
 - Provides a software toolkit to archive, process, display, and analyze standard event log messages
 - Provide mission operations tool that is highly portable across platforms and applicable to multiple missions
- Required no changes to other applications
- Moderate development effort (.5 FTE – first version)

The screenshot shows the GREAT software window with a menu bar (File, View, Help) and a toolbar. Below the toolbar are dropdown menus for font (Tahoma), size (11), and style (Normal). The main area displays a table of event log messages.

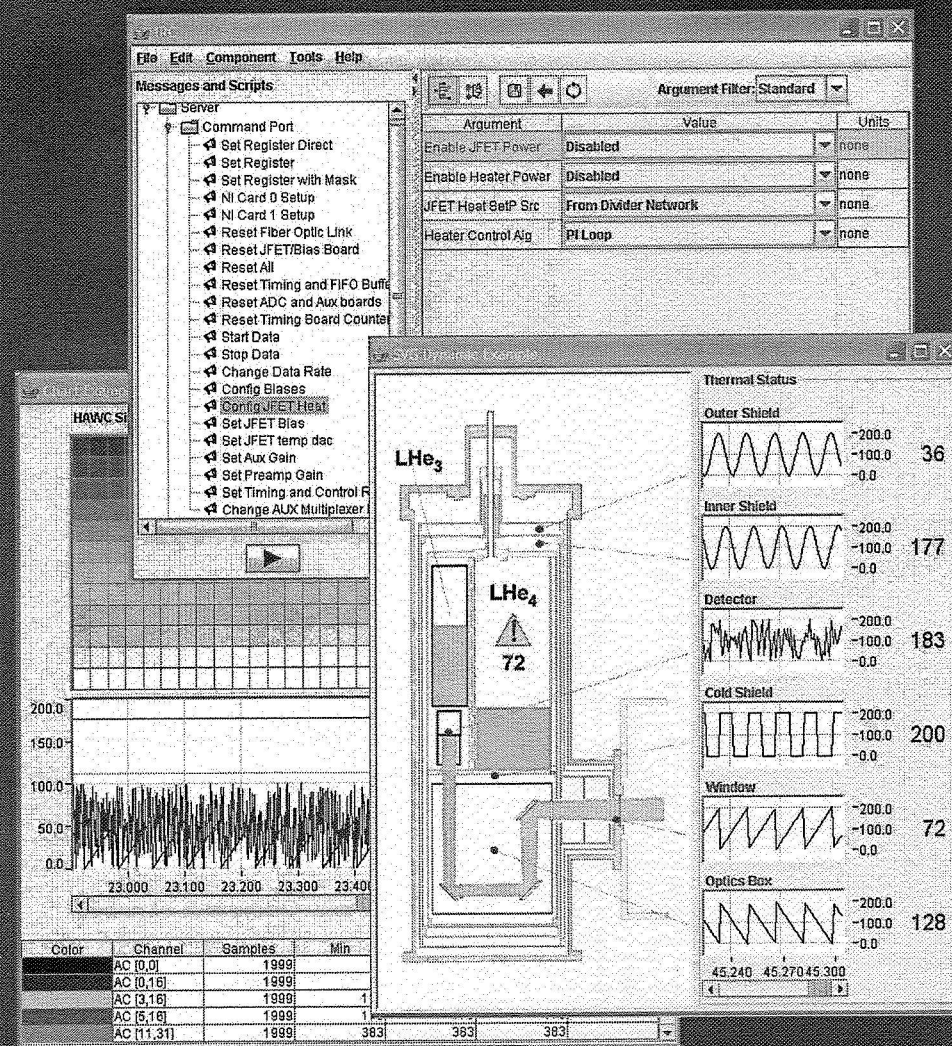
Time	ID	Type	Node	Message
2002/332/00:01:48.3619	30084	CMS	vibc01	Detected 0 miscomp out of 52 for config mon set JWFHRO...
2002/332/00:01:48.3637	30084	CMS	vibc01	Detected 0 miscomp out of 21 for config mon set KHOLD5M.
2002/332/00:01:48.3666	30084	CMS	vibc01	Detected 0 miscomp out of 63 for config mon set MNORMAL...
2002/332/00:01:48.3681	30080	CMS	vibc01	Miscomp ^NSBITERR = 1,000000; expected = EQ 000000...
2002/332/00:01:48.3695	30082	CMS	vibc01	Detected 1 miscomp out of 35 for config mon set NOP.
2002/332/00:01:48.3718	30084	CMS	vibc01	Detected 0 miscomp out of 39 for config mon set OOP2.
2002/332/00:01:48.3735	30084	CMS	vibc01	Detected 0 miscomp out of 18 for config mon set OTASTNRD.
2002/332/00:01:48.3753	30084	CMS	vibc01	Detected 0 miscomp out of 23 for config mon set UOP.
2002/332/00:04:54.4883	22410	CCL	vicc01	Op input (newton simon2 master): *XSRDSPB 3 1300 1 ...
2002/332/00:04:54.8669	22055	CCL	vicc01	***** Starting procedure XSRDSPB, *****
2002/332/00:04:54.8723	22034	PRC	vicc01	XSRDSPB: EV Version 0 06/05/01 Subsys=DMS...
2002/332/00:04:54.8764	22034	PRC	vicc01	XSRDSPB: EV WARNING! - This procedure cont...
2002/332/00:04:54.8802	22034	PRC	vicc01	XSRDSPB: global answer
2002/332/00:04:54.8836	22034	PRC	vicc01	XSRDSPB: EV PROC FOR LOADING & EXECUTI...
2002/332/00:04:54.8870	22034	PRC	vicc01	XSRDSPB: EV SSR. ONCE RECORDER P/B ARR...

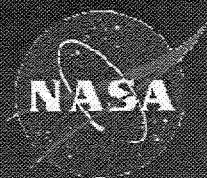
Below the first table is a second table with similar data. At the bottom of the window is a search bar with a magnifying glass icon and the text "Search".



Tools modified for GMSEC – Instrument Remote Control (IRC)

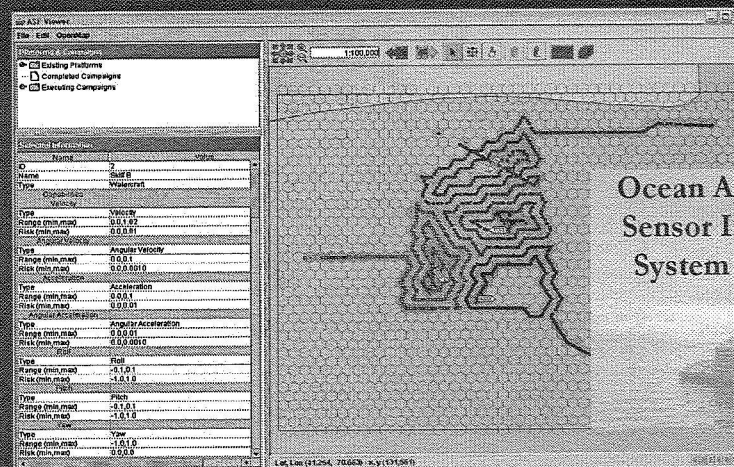
- Bridges GMSEC Bus to devices and sensors or Users
 - Enables integration of non-GMSEC devices via Instrument Markup Language (IML) to GMSEC system architectures
 - Enables customized Graphical User Interfaces to GMSEC enabled applications or devices
 - Can be used as an application framework for building customized GMSEC aware client or service applications
- Required no changes to other applications or device interfaces
- Extremely small development effort (.1 FTE)



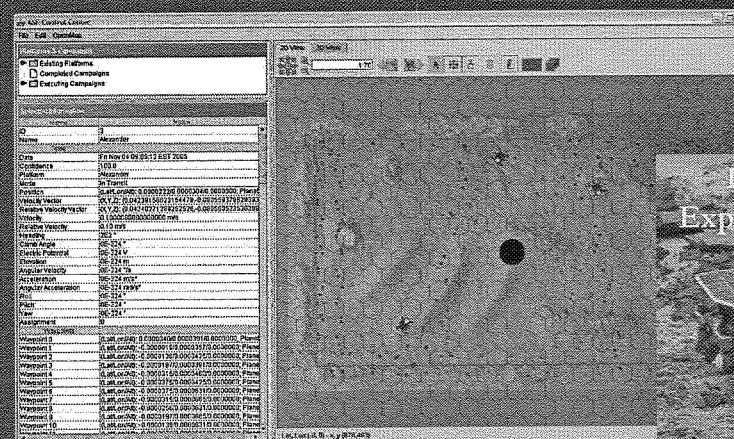


Tools modified for GMSEC – Adaptive Sensor Fleet (ASF)

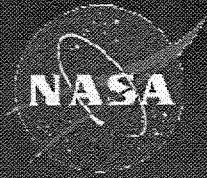
- Supervisory control system designed to use a collection of heterogeneous robotic platforms to optimally perform observations of dynamic environments driven by high-level goals.
 - provides for observations through high-level goals,
 - supervisory fleet management of robotic platforms (coordination),
 - analysis of environmental science data to use in the decision making process (collaboration),
 - optimal path planning and replanning,
 - identification of science phenomena,
 - and adaptation to dynamic or unknown environments Required no changes to other applications.
- **Required no changes to other applications**
- Adapted for GMSEC with 0.1 FTE (approx. 1 month)



Ocean Atmosphere Sensor Integration System (OASIS)



Rover Experiments



Summary

- Standardizing interfaces (API and messages) results in
 - Reduced programming integration time
 - Reduced system integration time
- Smart distributed architecture design enables
 - Quick, rapid, and cheap development of applications for targeted problems and/or system-wide areas.
 - High return on investment by using the strengths of your development teams as far as platform, language, and product
 - Independent development of new applications with minimal, or no, impact on other applications and their development teams
 - Added value to system architects by providing a broad suite of tools and applications