Onboard Plug and Play Standardization Effort
Software System Engineering & Technology Infusion Group
Overview
Amalaye Oyake, NASA/JPL

Introduction

This work introduces "Onboard Plug and Play Standardization".

This work is a study of the role of plug and play services for spacecraft.

Idea:

• There exist use cases where plug and play services would give system engineers greater leverage in spacecraft development and automation.

• A standard Plug and Play architecture is envisaged as mission complexity grows (Mars surface ops/Constellation missions etc).
Agenda

- Task Summary
- Task Overview
- Success Criteria
- Planned Products
- Technical Approach
- Issues and Concerns

Task Summary

Objectives:
- Explore uses for Plug and Play in space operations and identify possible JPL based contributions, to support standards creation.
- Recommend set of standards after a thorough search of the solution space (desktop, commercial).
- An abstract representation of the solution using a modeling language (UML, IDL).

Approach:
- Study commercial standards (WDM, Fieldbus, UPnP)
- Review current JPL practices and develop SOIS use cases.
- Develop JPL Mission specific and broader requirements - generate device attribute list.
- Develop recommendations to CCSDS WG.

Team:
- Amalaye Oyake, Lead Engineer

Schedule:
Task Overview: Importance of this Work

This work is important for several reasons:

- **Spacecraft complexity** is increasing with new missions and goals.

- **Lack of interoperability standards** is an obstacle to future inter-agency cooperation.

- A **Plug and Play standard** will benefit other services such as Monitor and Control.

Task Overview: Impetus or this Task

1. The growing need (for NASA) to facilitate spacecraft interoperability with software standards that leverage current industry trend.

2. The goals of today's space and Earth science programs call for many scientifically compelling, exciting missions, however much of the effort and cost for these missions is slowed down by increased complexity in software design, legacy hardware, and software hardware integration.

3. **Plug and Play** will help onboard services interoperate (missions/test/integration/verification/validation).
Task Overview: Importance of this Work

- Decouple the user application (software) from the spacecraft hardware
  - At device level (sensors/actuators) ~ Device proxy patterns.
  - At application layer ~ Other proxy patterns (separate emphasis)

- Use hardware with different data bus types
  - Changing the data bus should only affect the interface, and not the application.

- Plug and Play at the device level will facilitate the migration of engineering sensors and actuators from dedicated wires onto a data bus.

Success Criteria

- A Proposal - Well defined device level Plug and Play use cases for SOIS.
- Commitment from several sectors of JPL. Early adopters.
- Positive acceptance from CCSDS, with follow-on work for a prototype implementation. International Standard (CCSDS).
Planned Products

- An abstract specification/reference design (UML/XML Schema). It should be represented conceptually without any hard software bindings – Peter Gluck

  This conceptual representation should the model is
  - Distributed points of control
  - Allowed associations
  - Roles and responsibilities
  - Definition of all the actors (see UML)

- A generic device attributes list, with prototype attributes for some instruments (star tracker, camera etc).

- Definition of device classes and device proxy types.

- Prototype and further funding.

Task Overview: Spacecraft Hardware Complexity

Many instruments ~ many buses ~ many challenges

- 1553
- RS232
- RS485
- VME
- Compact PCI
- Custom designs
Use Cases

Ongoing discussion on plug and play concepts for both hardware devices and software applications:

- Test and integration
- Instrument coming alive
- Application level plug & play
- Device level plug and play
- Cooperating space craft
- Rovers Interacting
- Sensor networks
- Spacecraft networks
Use Cases: Surface Operations

Use Cases: Formation Flying
Technical Approach: Previous Efforts at JPL

- SuperMoca – Demo used Fieldbus FMS
  JPL - sponsored task aimed at developing an architecture and a set of technologies and associated open system specifications for the monitor and control of remote space vehicles and supporting remote ground networks.
  Status: No longer ongoing ~ Michael K. Jones.

- SIM/MAM/FIT Testbed – Uses CORBA (ACE-TAO) as the orb, a CORBA centric solution. Robust device proxy implementation with devices supported as C++ Objects.
  Status: Fully implemented, ground based.
  Actually a good reference model.

Technical Approach: External Initiatives

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
<th>Organization</th>
<th>Age</th>
<th>Physical Layer</th>
<th>Speed</th>
<th>Plug and Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 1451</td>
<td>Fieldbus interface standard</td>
<td>IEEE</td>
<td>1997</td>
<td>NCAP dependent</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Foundation Fieldbus</td>
<td>Fieldbus interface standard</td>
<td>Foundation</td>
<td>1995</td>
<td>Ethernet</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>UDPv6</td>
<td>Universal Plug and Play Standard</td>
<td>Open/MS</td>
<td>1993</td>
<td>Ethernet</td>
<td>Ethernet Speeds</td>
<td>YES</td>
</tr>
<tr>
<td>ARINC 653</td>
<td>Airborne Standard</td>
<td>ARINC</td>
<td>1997</td>
<td>1553</td>
<td>1553 Speeds</td>
<td>YES</td>
</tr>
<tr>
<td>SensorML</td>
<td>Sensor Markup Language</td>
<td>OpenGIS</td>
<td></td>
<td>No Bus Architecture</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>ASAPL</td>
<td>Architecture for Automotive modeling</td>
<td>Society of Automotive Engineers (SAE)</td>
<td>1991</td>
<td>Based on HC-11</td>
<td>Not bound to any Bus Architecture</td>
<td>Unknown</td>
</tr>
<tr>
<td>Jini</td>
<td>Sun device communication protocol</td>
<td>Sun Microsystems</td>
<td>1996</td>
<td>Ethernet</td>
<td>Ethernet Speeds</td>
<td>YES</td>
</tr>
<tr>
<td>BlueTooth</td>
<td>Bluetooth wireless</td>
<td>Bluetooth Wireless</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10/24/2003
Technical Approach – Opinions

- Peter Glück - Make framework abstract
- Kim Gostelo - Hardware Abstraction Layer
- Roger Klemm - Model a real system (DS1) with UML
- Vince Randolph - Take note of REAL technical obstacles
- Lee Eccles - Advice on Electronic Datasheets

Technical Approach: The Need for Standards

- The OSI model defines 7 layers and provides a good reference point for an Onboard Spacecraft Monitor and Control Standards model.
- Using this as a reference, our work will focus on the Presentation and Application layers only. Need compatibility across various physical and datalink layers.
Technical Approach: Case Study ~ Fieldbus Messages

Fieldbus Messages

Fieldbus Access Manager

Data Link Layer

Physical Layer

User Application

Fieldbus Message Specification

User Data

Protocol Control Information

Protocol Data Unit

Frame Check Sequence

Protocol Control Information

Protocol Data Unit

There may be more than one octet of preamble if repeaters are used.

Engineering View

Application

ASAP

Application Layer

Onboard Services

TSAP

Network Layer

Transport Layer

Data Link Layer

Physical Layer

Spacecraft Bus

1553/RSXXX/VME

Locally Connected Sensors

10/24/2003

10/24/2003
Technical Approach: SOIS Hardware Support Wish List

Data / Control Bus
- MIL-1553B Data Bus (from NASA-GSFC, NASA-JPL, Honeywell, CNES)
- 4-255 Data Bus (from ESA)
- MIL-1773 Data Bus (from NASA-JPL)
- (10/100/Gigabit) Ethernet (from NASA-JPL, Fisher-Rosemount, SRI International)
- VME/Compact-PCI (from NASA)

Serial Link
- Spacewire (JPL/ESA)
- Liaisons Numeriques Rapides (LNR) (from CNES)
- IEEE-1394 (from NASA-JPL, NASA-GSFC, Honeywell)
- Packet Wire (from Saab Ericsson Space)
- Inter-IC (I2C) Bus (from NASA-JPL, NASA-GSFC, Honeywell)
- Foundation H1 Fieldbus (Boeing, Fisher-Rosemount, SRI International)
- RS232, RS4XX (Legacy applications)

Discrete Interfaces
- 16-bit Serial Digital Output Interface
- 16-bit Serial Digital Input Interface
- 16-bit Bi-directional Serial Digital I/F
- Bi-level Discrete Input Interface
- Analogue Signal Interface
- On/Off & High Power Pulse CMD I/F
- Switch Status Interface
- Timing and Clock Output Interface (All Discrete I/Fs from ESA)

Spacecraft Internal Data Rates

10/24/2003
Technical Approach: Represent Layers With UML Subsystem Diagrams

Technical Approach

- Study the CCSDS requirements and M&C Layers - Done
- Case studies - Study current ESA/JPL practices - Done
- Hardware Abstract Layers, device driver models, design patterns of device drivers - Done
- Study WDM Model/Apple IO Kit/Fieldbus ... as a reference point - Done
- Current hardware standards/trends (JPL/Industry) - Done
- Consult domain experts and tabulate a device attribute list - Done
- Develop abstract representation for the concepts (UML) - Ongoing/Refinement
- Write a proposal for an actual implementation - In progress.
Issues and Concerns

- The level of buy-in by JPL and related agencies needs to be discovered and the solution space within JPL needs to be defined.

- These concepts need to be marketed, within JPL. CCSDS must address the propagation through SOIS, MOIMS, NASA, ESA/etc - ONGOING

- Once interest is generated, there need to be "birds of a feather" meetings, working groups - ONGOING

- Besides NASA/ESA adoption, SHOULD these ideas need to introduced to the commercial vendor domain?

Conclusions

- Need to work in the domain of space qualified systems*. 

- Most space devices are dumb
  - The device layer needs to implement good device proxy architecture that exposes device metadata and fully encapsulates concepts.

- Formalize data "on-the-wire"
  - Need to learn about PUS ...
  - Onboard data formats
  - OMG/CCSDS/ESA
  - Proprietary
  - Fieldbus
  - XTCE

- For device driver layer
  - Need mature design patterns that.
  - Driver models that can be layered.
Conclusions

- Depict high-level overview of requirements
- Formalize onboard subsystem software interfaces.
- Depict high-level overview of design
  - Use UML package/subsystem diagrams
- Logically factor the complete systems into subsystems
  - Use UML subsystem diagrams
  - Use UML interfaces to specify interconnects

Monitor and Control is fundamentally about data-representation.

We are primarily concerned with Metadata. Specifically and extensible metadata model, to represent commands, events, units and other interesting data.

Propose: Prototype model using JPL ALAB, RTC Framework (XML - Specifically SensorML, and capturing IEEE 1451 type data attributes which has been proposed before - Micah Clark/R&TD/2002, MTS over Spacewire).
Technical Approach: SOIS Hardware Support Wish List
What Do These Technologies Buy JPL?

- A Device Attribute List that we can reference
- A Device Messaging Specification
- A Pattern for device commanding
- A successful framework implementation

### Device Description Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANUFACTURER</td>
<td>Identifies the device manufacturer.</td>
</tr>
<tr>
<td>DEVICE_TYPE</td>
<td>Identifies the specific device.</td>
</tr>
<tr>
<td>DEVICE_REVISION</td>
<td>Revision of the device.</td>
</tr>
<tr>
<td>REVISION</td>
<td>Revision of the Device Description.</td>
</tr>
<tr>
<td>VARIABLES</td>
<td>An item of data.</td>
</tr>
<tr>
<td>MENU</td>
<td>A menu, listing other menus or displays for selection.</td>
</tr>
<tr>
<td>CHANGEABLE_VARS</td>
<td>A display for the operator to edit variables.</td>
</tr>
<tr>
<td>METHODS</td>
<td>An operational procedure for interaction with the field device.</td>
</tr>
<tr>
<td>ARRAYS</td>
<td>An array of variables of similar type.</td>
</tr>
<tr>
<td>RECORDS</td>
<td>A data structure of variables of mixed type.</td>
</tr>
<tr>
<td>ITEM_ARRAY</td>
<td>An array of items of similar type.</td>
</tr>
<tr>
<td>COLLECTION_OF</td>
<td>A collection of items of similar type.</td>
</tr>
<tr>
<td>VARIABLE_LIST</td>
<td>A list of variables, arrays or records.</td>
</tr>
<tr>
<td>BLOCK</td>
<td>Defines a Function Block (or other block) by listing its contents.</td>
</tr>
<tr>
<td>REFRESH</td>
<td>A set of variables whose values must be re-read whenever other specified variables are changed.</td>
</tr>
<tr>
<td>UNIT</td>
<td>Defines which variables use a particular engineering unit.</td>
</tr>
<tr>
<td>WRITE_AS_ONE</td>
<td>A set of variables which must always be written as a group.</td>
</tr>
<tr>
<td>PROGRAMS</td>
<td>An executable program in the field device.</td>
</tr>
<tr>
<td>DOMAIN</td>
<td>A memory area in the field device.</td>
</tr>
<tr>
<td>RESPONSE_CODES</td>
<td>A set of possible response code values and their meanings.</td>
</tr>
</tbody>
</table>

10/24/2003
Attributes of Interest

- Classes
  - Sensors, Actuators
- Subsystem
  - ACS
  - TEL
  - CDS
- Types (Smart, Board, I/F)
  - IMU, SRU, SS, CAM, SDST

List of OUPUTS/INPUTS
- MaxLatency
- Data Size
- Data Format
- Data Type
- Data Rates
- Precision
- Accuracy
- Cycle Sync.
- "Start Now"

Start/Stop Talk
- Uncertainty
  - Jitter
- Calibration
- Sequencing
  - How
  - What
  - Procedure

33 10/24/2003

Attributes of Interest

- Delay Between Commands (min/max)/TLM
  - Heartbeat
- Time Synchronization
  - Freq FWD/REV
  - Precision
  - Accuracy
  - Format
- Status Messages
  - On/Off
  - Standby/Operate
  - Offline/Online
  - Busy/Ready
  - Loading (SW)
  - Boot/Recycle

Performance Metrics
- Made/Missed Deadline
- Excess Processing Available
- Avg. or Expected processing duration estimate
- Timestamp
- Dump

Dependencies/Associations

Resource Requirement

Resource Availability
- Power
- Memory
- Processing Cycles
- Mass Properties
- Thermal Properties
- Vibration/Dynamics
- Radiation Tolerance
- Bright Body Constraints

34 10/24/2003
Attributes of Interest

- **Resource Availability (Cont)**
  - FOV
    - Exclusion
    - Zone
    - Duration
    - Rate
  - Consumables/Consumability
    - Max Operating Time
    - Warm-Up Time
- **Data Buses**
  - 1553
  - RS422
  - I2C
  - Wi-Fi (802.11a/b/g)
  - Bluetooth
  - 1394a/b
  - USB
- **Protocols**
  - TCP/IP
  - UDP
  - Proximity-1
  - CCSDS
  - Plug and Play
- **Care & Feeding Instructions**
  - Procedures for Resetting DEV
  - Flight Rules
  - Fault Recovery
  - Handling Instructions
  - Requests (Responses) Def.
  - Recovery Instructions
- **Data Types**
  - SI Units ***
  - Science
  - Health
  - Engineering
  - Status
  - Warnings/Errors