Using XML and Java Technologies for Astronomical Instrument Control

Troy J. Ames (Troy.J.Ames@gsfc.nasa.gov),
NASA Goddard Space Flight Center

Lynne Case (Lynne.Case@commerceone.com),
Commerce One

NOTE: Derived from prior work of Troy Ames, NASA/Goddard Space Flight Center and Ken Sall, Craig Warsaw, and Lisa Koons of Commerce One
Agenda

- Project Background
- Technologies Used
- XML Uses and Examples
- Framework Architecture
- Benefits of the Architecture
IRC Project Goals

• Instrument Control Software
  – Extensible framework for the distributed control and monitoring of remote instruments
    • Control = commanding of the instrument
    • Monitoring includes quick-look visualizations of data
  – Promote reuse by design
    • Reuse = use for controlling multiple kinds of instruments
  – Easy to develop, modify, maintain, and extend
  – Platform independent

• Develop reliable, robust instruments
  – Easy for scientists to use
  – Clear interface between hardware and software
  – Support for iterative development
Instrument Remote Control Project (IRC)

- NASA Goddard Space Flight Center (GSFC)
  - Advanced Architectures and Automation Branch (Code 588)
  - Infrared Astrophysics Branch
- Commerce One
- Center for Astrophysical Research in Antarctica (CARA)
  - University of Chicago, Yerkes Observatory
- Stratospheric Observatory for Infrared Astronomy (SOFIA): NASA, USRA, DLR
  - HAWC: High-resolution Airborne Wideband Camera
  - SAFIRE: Submillimeter And Far Infrared Experiment
Technology Used in IRC

• Java™ Technology
  – High productivity rate for development
  – Support for reusable components
  – Rich set of APIs
    • Networking
    • Graphics
    • GUI
    • Security
  – Platform independence
  – Many tools available

• XML – Extensible Markup Language
  – Metalanguage -- a language for describing other languages
    • Document Type Definition (DTD) or schema defines specific dialect
  – Structured, hierarchical data
    • Human readable
    • Machine-understandable
  – Platform independent
  – Many tools available
XML Uses In IRC Framework

• Instrument Markup Language (IML) - to describe instruments
  – Logical command set
  – Command arguments, constraints, and units
  – Field data types and valid values/ranges
  – Logical data streams (telemetry)
  – Command and Data formats
  – Communication mechanisms

• Pipeline Algorithm Markup Language – to describe data analysis pipeline algorithms
  – Reuses many of the constructs in IML
IML History

• First IML Prototype DTD (Feb '99)
  – HAWC Simulator
  – ASCII commands
  – Simple telemetry
  – One port type (TCP)

• Engineering Test DTD (Fall '99)
  – Detector downselect candidate for SPIRE
  – Binary commands
  – Complex telemetry with parsing delegate
  – Several port types (DMA, Serial, TCP)

• XML Schema (currently testing)
High-Level IML File Structure

<Instrument name="HAWC" ... />
<Instrument name="Telescope" >
  <CommandInterface name="Telescope Commands">
    <Command ... />
  </CommandInterface>
  <DataInterface name="Telescope Status" >
    <DataElement ... />
  </DataInterface>
</Instrument>

<Port name="Telescope Command Port" portType="TCP" >
  <Format name="Command Format" formatType="command" .../>
  <InterfaceReference reference="Telescope Commands" />
</Port>

<Port name="Telescope Status Port" portType="Serial" >
  <Format name="Status Format" formatType="data" ... />
  <InterfaceReference reference="Telescope Status" />
</Port>
</Instrument>
Command Example

<Command name="Move" >
  <Field name="RA" type="Sexagesimal" >
    <RangeConstraint low="00:00:00.0" high="23:59:59.99"
  />
</Field>

<Field name="DEC" type="Sexagesimal" >
  <RangeConstraint low="-89:59:59.99" high="89:59:59.99"

  />
</Field>

<Field name="Epoch" type="Float" />
</Command>
IRC Device Configuration

Public Interfaces

IRC Device

Private Interfaces

Sensor, Actuator, Hardware Device, or IRC Device

IML

PAML
Example Device Internal Configuration

- This is an example internal configuration of an IRC device.
- Commands and responses typically flow through one port while the data flows through separate ports.
- An internal pipeline might provide data through one data port while the System Model might route interim data through a second data port.
Example Detailed Device Internals
IRC Project Status

- Demonstrated feasibility of approach
- First engineering test of instrument complete
- Currently testing an enhanced IRC framework
  - IML conversion to XML Schema
  - IRC Device for distribution architecture
  - Easier algorithm development
  - More generic algorithms provided
- Applying IRC
  - HAWC
  - FIBRE (a prototype of SAFIRE)
  - SAFIRE
The Vision: Instrument Design

- Hardware Engineer defines instrument
  - Uses custom editor to hide XML details
- Data Analysis Pipeline
  - Select from library of algorithms
  - New algorithms can be created by instrument designer or end user (using Java, scripts, native code, IDL, …)
  - Create and save baseline pipeline configurations
- Graphical User Interface
  - Default GUI may be sufficient for engineering testing
  - In general, need to be able to customize GUI
- Define calibration and other scripts
The Vision: Value Proposition

- Generic architecture
- Driven by descriptions
- Significant code reuse
- Anticipate 10% to 30% customized code
- Savings of 70% to 90% over traditional development paradigms
For More Information

- NASA/Goddard IRC website (papers, presentations, DTD):
- NASA/Goddard XML for Astronomy website
- Listed on XML.org
  - http://www.xml.org/ - both DTD and XML Schema listed
- Contact:
  - Troy Ames: Troy.Ames@gsfc.nasa.gov, (301) 286-5673
  - Julie Breed: Julia.Breed@gsfc.nasa.gov, (301) 286-5049