Atacama Large Millimeter Array

ACS Common Software for the ALMA Project

G.Chiozzi, B.Gustafsson, B.Jeram, P.Sivera – ESO
M.Plesko, M.Sekiranja, G.Tkacik, J.Dovc, M.Kanduc, G.Milcinski, I.Verstovsek, K.Zagar - JSI

ICALEPCS 2001 Nov. 27-30, 2001
Contents

- What is ALMA? ➡ THAT004 - M. Pokorny
- Why a common software?
- What are the key ideas in ACS?
- What is the Architecture of ACS
- How are we going to go on?
- How can you get more information
ALMA Common Software (ACS)

ALMA project is highly distributed: many sites and many “development cultures”

- ACS aims at providing an answer to the following needs:
  - common full framework, not only few libraries (for a very distributed group) (e.g. EPICS).
  - well tested software that avoids duplication.
  - make upgrades and maintenance reasonable.
  - incremental development via Releases.
  - standardisation offering environment with design patterns (rather than rules).
  - common configuration control/installation procedures.
    It worked on a comparable size project: the VLT.
ACS start-up: Summer 1999

- Preliminary discussions on ACS objectives and structure:
  - VLT experience
  - New technology: CORBA
- Decided to start from existing product, to make use of CORBA experience.
- Collaboration with JSI-Ljubliana to further develop existing system built for High Energy Physics (ICALEPCS ‘99, Trieste).
ACS Architecture

1 - Base tools
- Development tools
- CORBA Middleware
- ACE
- Device Drivers

2 - Core components
- Distributed Object
- Data channel
- Error System
- Logging System
- Time System
- Astro libraries

3 - Services
- Management and Access Control
- Archiving System
- Command System
- Alarm System
- Sampling

4 - Hi-level APIs and tools
- UIF libraries
- Scripting
- ACS Application Framework
- FITS libraries

Applications

...more to come...
**DO-Property-Characteristics**

- **DO**: base class for any physical/logical Device (e.g. temperature sensor, motor)
- Each DO has **Properties** (e.g. status value, position - control/monitor points)
- **Characteristics** of Dos and Properties (Static data in Configuration DB, e.g. units, ranges, default values)
UIF Libraries

Kitt Peak test (Dec.00):
Pointing/tracking with 12 m Radio + optical tel.
ACS Status

- Oct. 2000: ACS v.0.0 (Prototype)
- Dec. 2000: KP test
- Sep. 2001: ACS Architecture Document
- Sep. 2001: ACS 1.0
  In use in Test Interferometer Control Software (TICS) (THAT004) and Test Correlator software
- Feb. 2002: ACS 1.1 (support for TICS)
- Sep. 2002: ACS 2.0

We plan an incremental release every 6 months
Supported Platforms

- Operating system: Linux, SUN OS, (MS-Windows)
- Real-time: VME, VxWorks (Phase1), CAN bus
- Languages: C++, JAVA, Python, (TCL)
- CORBA middleware: TAO (& ACE) (C++), Orbacus (Java), Omniorb (Python), CORBA services.
- Unified Modeling Language (UML)
- DB2 being evaluated as configuration DB
ALMA Sites
Chajnantor

www.eso.org/projects/alma
www.alma.nrao.edu/development/computing
www.eso.org/~gchiozzi/AlmaAcs
http://kgb.ijs.si/KGB/

ICALEPCS 2001
THAT004: TICS
THCT004: CSML
THAP005: CORBA
THAP025: Abeans
Conclusion

- Developed keeping in mind the needs of astronomical and accelerator control projects.
- Can easily run on many platforms.
- Open source (GPL licence).
- Free development tools and ORBs.

We think that many other projects can use ACS. A wider user’s base can provide valuable feedback.
Extra slides

The following pages contain some extra slides that we do not plan to show at the conference, but that could be useful.
Data Channel
Logging

Logs are sent to the central server when a low frequency timeout expires or the buffer is full. Messages are filtered to compress identical messages generated in a time period.

Subject
(from Data channel)

Observer
(from Data channel)

Update()

Detach()

Notify()

(send logs to central ser...)

producer for logs

client for logs

send logs

logs are sent to the central server when a low frequency timeout expires or the buffer is full. Messages are filtered to compress identical messages generated in a time period.

Data Channel

Subscribes to logs

Un-subscribe to logs

0..n

Subject
(from Data channel)

Attach()

Detach()

Notify()

0..n

logMangerProxy

buffer

ACS_LOG()
Management and Access Control

ALMA Project

ALMA Common Software
Archiving System

Relational Database

store/retrieve data

logMonitor
(from Logging System)

archiveManager

browse logs

browse monitor data

producer of monitor data

producer for logs
(from Logging System)

monitorDisplay

through data channel

through data channel
OE and Admin Client
Major effort in documentation

- Introductory documents (Overview, Installation, Rel Notes)
- Specification Documents
- User Manuals and Tutorials
- Abeans
- ACS IDL Online documentation
ACS 1.0 Documentation - 2

- ACS API Online Documentation
- ACS ABeans Online Documentation
- Useful VLTSW Documents

See:
- ACS_1_0_Linux-CD/Docs
- $ACSROOT/man (online reference, html and man)

ACS 1.0 ABeans Online Documentation

ACS ABeans is the online documentation of the ACS 1.0 ABeans Java class library. It has been produced with Doxygen with from the actual code. Available also as a printable reference manual in pdf format (1.5MB) WARNING: This is a very big document and Netscape or MS Internet Explorer can have performance problems in displaying the main tree.

ACS 1.0 Object Explorer API online documentation

This is the online documentation for the ACS 1.0 Object Explorer classes. It has been produced with Doxygen with from the actual code. Available also as a printable reference manual in pdf format (290KB).

Useful VLTSW Documents

This section contains manuals and documentation for tools that, although part of the ESO VLT Common Software, are distributed as integral part of ACS 1.0.

- Solaris Installation Manual - VLT-MAN-ESO-17200-2164 (pdf, 23KB)
- Configuration Management Module User Manual (CMU) - VLT-MAN-ESO-17200-0780 (pdf, 1MB)
- Installation Tool For VLTSW Packages User and Maintenance Manual (pkgman) - VLT-MAN-ESO-17200-1913 (pdf, 58KB)

Disclaimer: The information in these pages is for ALMA Common SW group internal use only.

Send comments to: chasity@eso.org
Last modified: Mon Sep 24 18:22:44 MET 2001
This is version: "4g(9)/$Id: index.html 1.18 2001/09/24 12:26:18 vltscm Exp $"
How much effort in ACS 1.0?

Source lines of code (for comments add about 10%) for:

- ACS cmm Modules (code developed for ACS)
- Java Abeans (mostly code inherited from ANKA)

<table>
<thead>
<tr>
<th>SLOC</th>
<th>Directory</th>
<th>SLOC-by-Language (Sorted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>61730</td>
<td>ACS</td>
<td>cpp=61536, sh=129, tcl=28, sed=23, csh=14</td>
</tr>
<tr>
<td>188820</td>
<td>Java</td>
<td>java=188820</td>
</tr>
</tbody>
</table>

- Total Physical Source Lines of Code = 252019
- Estimated Development Effort in Person-Years = 66.46
  (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
- Total Estimated Cost to Develop = $8977317
  (average salary = $56286/year, overhead = 2.4).

Data generated using 'SLOCCount' by David A. Wheeler
http://www.dwheeler.com/sloccount/sloccount.html

More than 2000 pages of printable design/reference documentation