# Open Source Software and its Role in Space Exploration

USENIX LISA 2006

December 6, 2006

djbyrne at jpl.nasa.gov

Jet Propulsion Laboratory

California Institute of Technology

JPL

# Common Goals

- FOSS (Free/Open Source Software) developers and NASA folks have a lot in common
  - Dedicated to expanding the pool of information floating freely through society
    - Space Act NASA Charter: "... provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof..."
    - http://www.nasa.gov/offices/ogc/about/space_act1.html
  - Focused on the cutting edge, creating tools and capabilities which did not previously exist
  - I like to think that FOSS community donations of code are reciprocated with knowledge about weather systems, climate, and basic science

JPL

# About the speaker

- Software geek
  - Whatever language
  - Whatever platform (well, I'm "done" with one...)
- SysAdmin
- System Engineer
- Programmer again!
  - As long as the SysEng stuff is done first


JPL

# Open Source in Space

- Explores our Solar System
- Observes the Universe
- Is used to develop new algorithms and code
- Is used to move and analyze data by flight operations on the ground

JPL

# The Open Advantage

- Faster
  - Procurement cycles alone... Oy!
  - Bug fix turn-around times, or we can do ‘em ourselves and give ‘em back
  - Feature additions - ditto, but we can only give back after a lot of paperwork (or contract for them)
- Reliability
  - We tend to find bugs which don’t bother other customers.  We live at or beyond the border cases
  - Full system visibility is key to characterization and resolution

JPL

# Open Advantage, cont

- Interoperability and Portability
  - Our industry, academic, and international partners can use their favorite platforms
  - Final production environments can be too scarce to pass around for development
  - Operational lifetimes can be decades on old platforms
- Openness
  - ITAR (International Traffic in Arms Regulations), EAR (Export Administration Regulations), and IP (Intellectual Property) are non-problems for existing Open code
    - ‘Though Adaptations and changes for mission details can be controlled and limited

JPL

# The Cost Question

- Cost of getting a product isn't a big factor
- TCO (Total Cost of Ownership) is dominated by learning curve, testing, reviews, writing procedures, etc.


JPL

# Confidence in the Future

- Developers like having the source code as a risk mitigator
- Managers like support contracts for the same reason
  - Glad to see companies offering these
- CMMI (Capability Maturity Model Integration) certification would help - puts a paper-trail on good practices like CM, test processes, etc.
  - Useless push-up or valuable scrub?

JPL

# NASA's Software Policy

- http://nodis.gsfc.nasa.gov/
  - Search for NPD 2820.1C
  - Some other good stuff there by searching for software
- Some relevant excerpts:
  - Require software providers (includes internal NASA providers) to have proven organizational capabilities and experience to deliver quality software on time, within budget, and within technical acceptability.
  - Require software providers to develop a plan to manage software throughout the program/project life cycle. This plan shall include the collection and reporting of actual software related expenditures at the project level by life cycle phases.
  - Release software in accordance with NPR 2210.1, External Release of NASA Software, consistent with law and applicable agreements, for commercial, industrial, educational, and governmental purposes.

JPL

# And now some examples

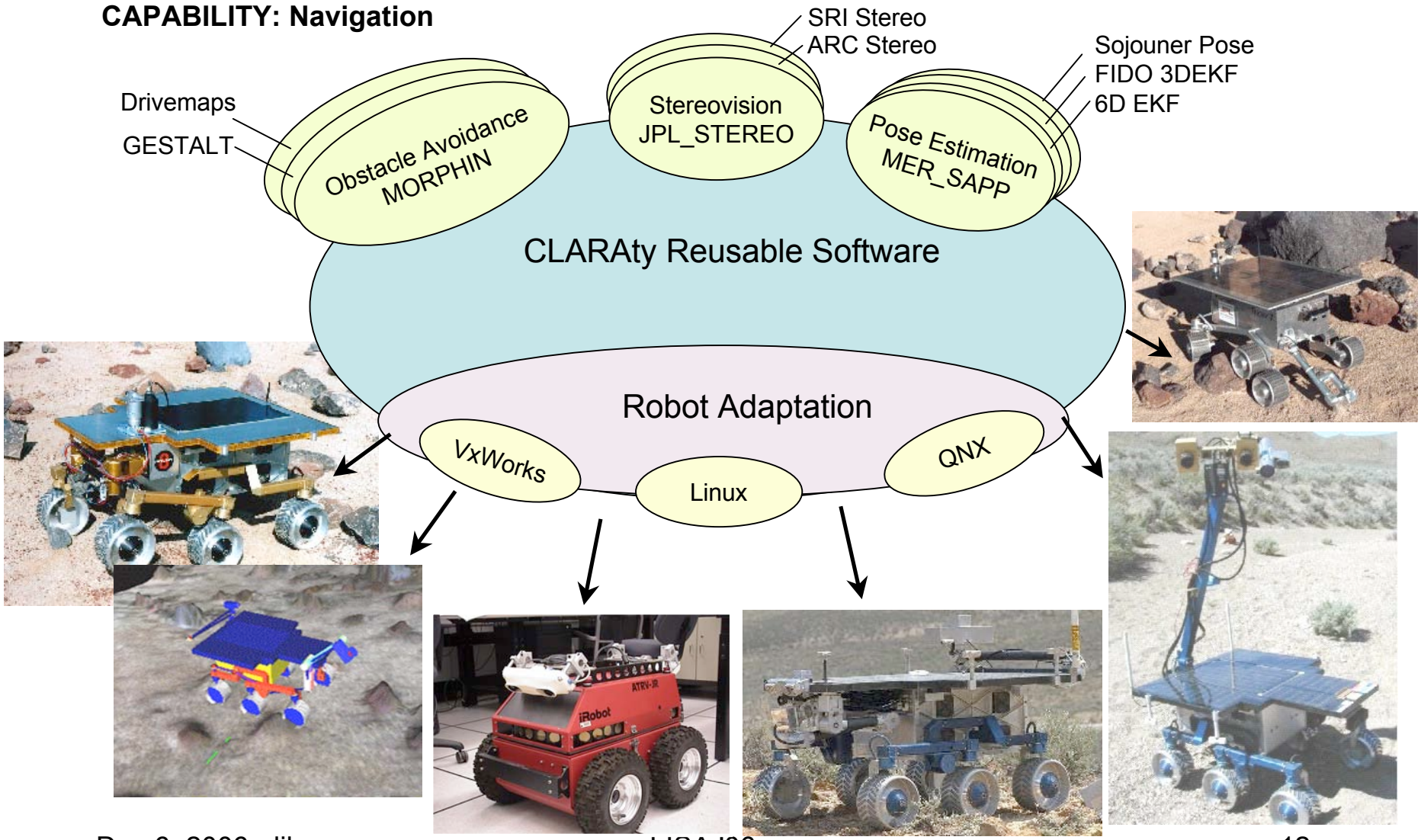
- CLARAty
- Electra
  - MRO Electra
  - MSL ElectraLite
  - MSL TDS
  - Chandrayaan-1 M3
- CCA
- Supercomputing
- Others

JPL

# CLARAty Project

- Coupled Layer Architecture for Robotic Autonomy
  - http://claraty.jpl.nasa.gov/
- Investigating robot visions, navigation, operator interfaces, simulation challenges, etc.
- Unified and reusable software that provides robotic functionality and simplifies the integration of new technologies on robotic platforms.
- Research tool designed for the development, validation, and maturation of various research technologies.

# Interoperability: Software & Hardware



JPL

# CLARAty: Key Challenges

- Robots have different physical characteristics
- Robots have different hardware architectures
- Contributions made by multiple institutions
- Advanced research requires a flexible framework
- Software must support various platforms
- Lack of common low-cost robotic platforms
- Software must be unrestricted and accessible (ITAR and IP)
- Software must integrate legacy code bases

JPL

# CLARAty examples

- Rovers, including next-generation for Mars, are being tested between many institutions
  - JPL, ARC, CMU, U-Minnesota, etc.
- Test images are shared and access-controlled via OpenAFS filesystem
  - Latest pix from Spirit & Opportunity are used; AFS of course keeps the test set the same for everyone
- Code in AFS, CVS (Concurrent Versioning System)
  - YaM (Rapid Software Development Framework)
- Ames rover w/ laptop under solar panel - running linux and AFS

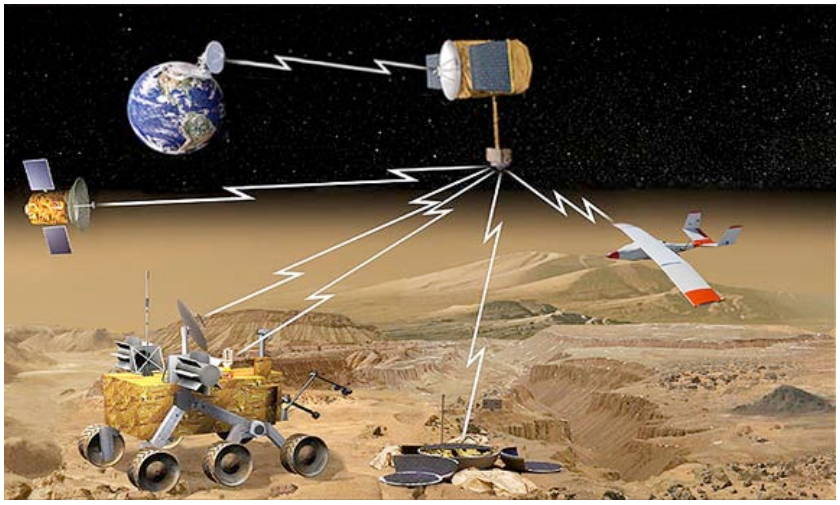
JPL

# CLARAty videos

- [26 seconds] ROAMS (Rover Modeling and Simulation) SOOPS (Science Operations on Planetary Surfaces)
- [63 seconds] GESTALT (Grid-based Estimation of Surface Traversability Applied to Local Terrain) on FIDO
- [41 seconds] SCIP (Single Cycle Instrument Placement)
- [180 seconds] SMC Rocky 7 style rover with reconfigurable wheel/arm (from Shigeo Hirose - Japan)

# Electra Radios

- NASA/JPL's product line of Software-Defined Radios (SDR) and spin-offs in support of Mars Network concepts, and InterPlanetary Internet



JPL

# Electra Radios, cont

- Provide UHF links in compliance with some CCSDS protocols
  - Consultative Committee for Space Data Systems
  - Proximity-1 (data link)
  - CFDP (file delivery)
  - http://www.ccsds.org/
  - http://www.ipnsig.org/

JPL

# Electra Radios, cont

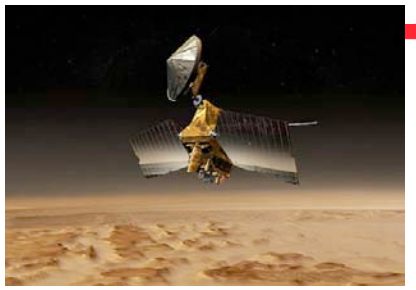
- Post-launch reconfigurability of protocol and signal processing functions
- Radio metric tracking for approach navigation, in situ surface positioning, and orbital rendezvous
- Timing services to support time synchronization of Mars exploration assets

JPL

# Electra Dev Environment

- Code lives in a CVS repository stored in AFS, using kerberos authentication
  - Mostly C
  - Some assembly for trap handling
- Cross-compiled on linux for RTEMS on SPARCv7 target
- Built with gcc, make, libtools, etc.
- Linked with newlib

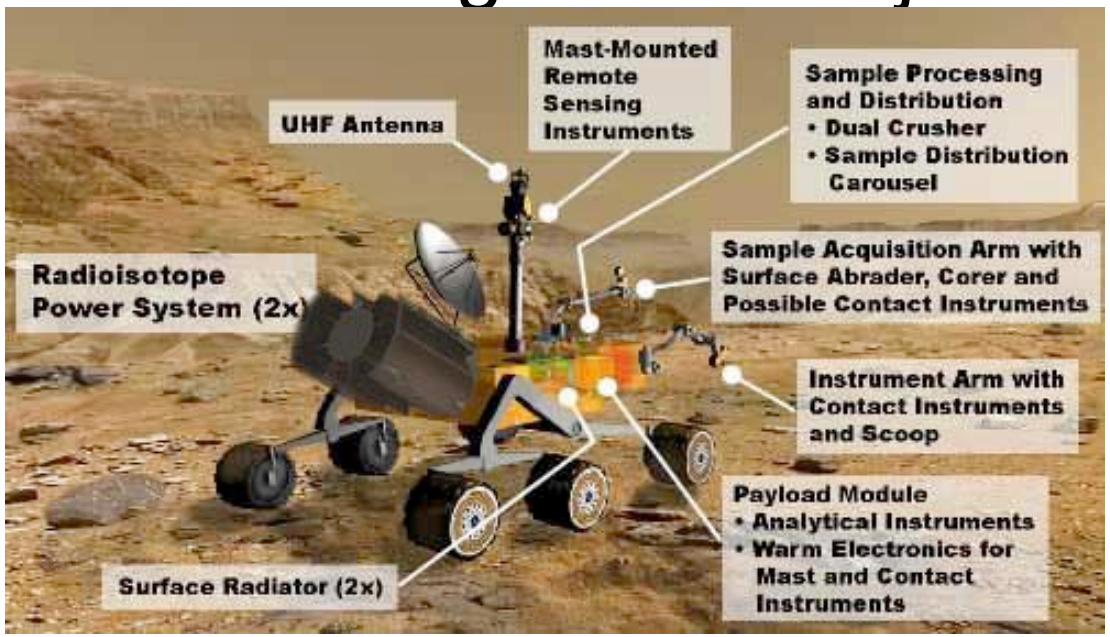
# Electra: MRO



- NASA/JPL's Mars Reconnaissance Orbiter
- Electra added as a payload for the Mars Network infrastructure, rather than used as the prime communications device
- Arrived at Mars in March, 2006
- Software fix for external radio interference, summer 2006
- Additional functionality will be needed for 2009 lander, MSL

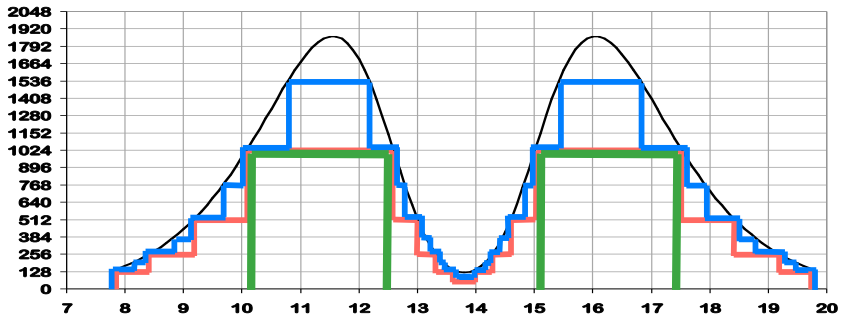
# MSL

- NASA/JPL's Mars Science Laboratory
- Mars rover to launch in 2009
- Much larger than Sojourner or MER




# MSL Electra-Lite

- Tighter mass/power constraints for lander than MRO
- Data throughput requirements drive new software radio function to adapt data rates during a communications pass by an orbiter
  - Easiest sequencing for either commanding or data return is single data rate, which needs margin
  - Adapting rates “fills in” lost bits under the optimum curve




**Example Data Rate Change Performance**

- 655 Mbits = Area Under the Black curve is Data Volume using a continuously variable data rate
- 521 Mbits = Area Under the Blue curve is Data Volume using a Root 2 Step Variable Data Rate
- 471 Mbits = Area Under the Red lines is Data Volume using a Factor of 2 Step Variable Data Rate
- 309 Mbits = Area Under the Green line is Data Volume returned using best Factor of 2 fixed data rate

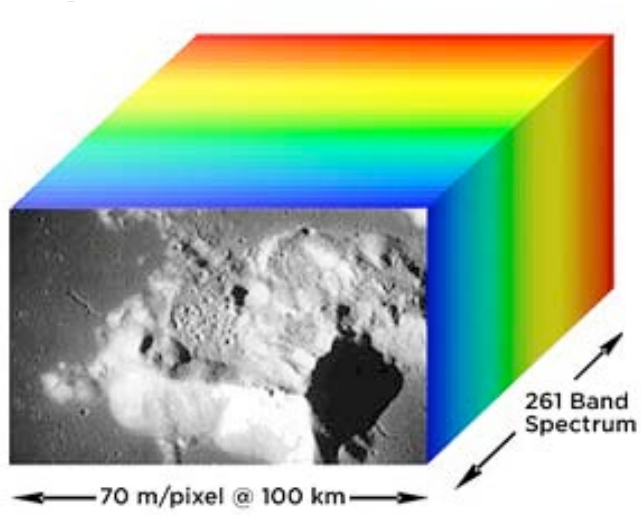
JPL

# MSL TDS

- Terminal Descent Sensor
- Landing RADAR
- Digital Electronics and software are Electra heritage
- [111 seconds] EDL (Entry, Descent, and Landing)

# Chandrayaan-1 M3

- ISRO's (Indian Space Research Organisation) Chandrayaan-1 is India's first moon mission, to launch in 2008
  - http://www.isro.org/chandrayaan-1/
- NASA is providing the Moon Mineralogy Mapper, a state-of-the-art im spectrometer instrument
- M3's spacecraft interface is again Electra heritage



JPL

# CCA

- NASA Constellation program's CCA (C3I (Command, Control, Communications, and Information) Communications Adaptor)
- Internet RFCs to be implemented on range of "space routers" for:
  - CEV (Crew Exploration Vehicle)
  - CLV (Crew Launch Vehicle)
  - LSAM (Lunar Surface Access Module)
  - EVA suits, Habitat, Lunar Rover, Lunar orbiters
- What terrestrial uses need delay or disruption tolerant implementations we could re-use?

JPL

# JPL Supercomputing

- http://sc.jpl.nasa.gov/
  - Aeronautical simulations, Lunar gravity map, Black Hole physics, electron collisions in plasmas - you know, stuff like that.
- Math libraries
  - LAPACK: A publically-available library (in source code) from Oak Ridge National Labs/University of Tennessee at Knoxville (http://www.netlib.org) which has become the standard for linear algebra solvers. Each of the manufacturers has created a version of this library optimized for their hardware.
- Beowulf clusters all over the place

# Where FOSS isn't found (yet?)

- Our business systems
  - E.g. timekeeping, benefits, staffing
  - The web-browser wars are alive and well.
- Management artifacts
  - E.g. presentation slides, schedules, budgets
  - OpenOffice sightings are rare
- Spreadsheet applications are becoming toolsets of their own

JPL

# Ubiquitous FOSS

- Operating Systems, Systems Management
  - Rocks (cluster linux), Ganglia, amanda
- Software Management
  - Depot, Subversion, Trac, Bugzilla
- Communications
  - OpenSSH, Apache, Jabber, Firefox/Mozilla, Sendmail, Mailman, Procmail, CUPS, OpenOffice, wikis (various)
- Data Visualization
  - ImageMagick, GMT, MatPlotLib
- Compilers, languages, code checkers
  - SunStudio, splint, Doxygen, valgrind
  - Java, Perl (some JPL history there), Python, Ruby
- DB
  - MySQL
- ...And on and on

JPL

# FItops Dev/OPS/project TPS listing

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| a2ps-4.13 | doxygen-1.4.2 | gdk-pixbuf-0.22 | ispell-3.2.06 | libxml++-2.10.0 | ogl13_rt32_64 | spin-4.1.2 | vnc-3.3.3r2 |
| Acrobat7.0.1 | dvdq-1.5 | gdome2-0.8.1 | j2sdk1.4.2_10-j | libxml2-2.6.19 | open_inventor-2 | sqlite-3.2.8 | Wcl-2.8rev4 |
| ActiveTcl8.4.11 | eclipse-3.1.1 | gettext-0.14.1 | j2sdkee1.3.1 | libxslt-1.1.6 | OpenOffice.org1 | startup-notific | webmacro-classi |
| afs3.6.2.53 | EditTable3.0.7 | gh-1.8.9-ppc | jaf-1.0.2 | linc-1.1.1 | openssh-4.2p1 | struts-1.2.7 | wget-1.5.3 |
| ammos_scripts_r | elm2.5.8 | gh-1.8.9-sol2 | jai-1_1_2 | lsof_4.75 | openssl-0.9.7d | subversion-1.1. | wind-1.0.1-ppc |
| antlr-2.7.1_exe | emacs-21.3 | gh-keys | jakarta-tomcat- | lynx2-8-5-16 | ossasn1.2159 | sudo-1.6.8p9 | wind-2.0-ppc |
| apache-ant-1.6. | enscript-1.6.1 | ghostscript-8.1 | javacc-4.0 | m4-1.4 | pango-1.4.0 | sunstudio11 | wind-2.0-vxsim |
| apache_1.3.29 | EPak3.0.7 | ghostview-1.5 | javamail-1.3.3_ | MagicDraw-7.5_f | pari-2.1.7 | SWIG-1.3.24 | wind-2.1-ppc-MG |
| aspell-0.50.5 | etjava61 | gimp-2.0.1 | jdk1.5.0_05_jav | make-3.81beta1 | patch-2.5.4 | sysinfo-5.0.0.1 | wind-2.2.1-ppc- |
| atk-1.6.0 | ets_diag-1 | gimp-print-4.2. | jedit4.2 | makedisc-4.21 | perceps-3.5.0 | tar-1.13.25 | wind-cmg-ppc-1. |
| autoconf-2.59 | expat-1.95.7 | glib-1.2.10fort | jflex-1.4.1 | marathon-0.90a | perl-5.8.7 | tcl8.4.6_thread | wxGTK-2.5.2 |
| automake-1.8.4 | expat-2.0.0 | glib-2.4.0 | jh2.0_02 | matlab-7.1sp3 | pfil-2.1.2 | tcp_wrappers_7. | wxMotif-2.2.7 |
| backupexpress-2 | expect-5.43 | glibmm-2.4.2 | jing-20030619 | mc-4.5.55 | pgp-2.6.2s | tcsh-sol8 | wxWindows-1.63e |
| batik-1.5.1 | FaultMeasuremen | glimpse-4.17.4 | jms3_6-plt | md5-rev1 | pine4.64 | teTeX-2.0.2 | x11-ssh-askpass |
| binutils-2.16.1 | fetchmail-6.2.5 | gnupg-1.2.4 | jmsproxy-1.3 | Mesa-6.2.1 | pkgconfig-0.15 | texinfo-4.2 | Xalan-C_1_9_0 |
| bison-2.1 | fftw-3.0.1 | gnuplot-4.0.0 | joe-3.1 | metamail-2.7 | plotutils-2.4.1 | textutils-2.1 | xalan-j_2_4_D1 |
| blt2.4z | fileutils-4.1 | gperf-3.0.1 | jpeg-6brev1 | mozilla-1.7.12 | psutils1.17 | thread-2.5.2 | xchat-2.0.9 |
| bxpro-5.0.4 | filter-2.5.1 | graphviz-2.6 | jswdk-2.3 | mpeg2dec-0.4.0 | ptplot5.3 | thunderbird-1.0 | xcursor-1.0.2 |
| bzip2-1.0.3 | firefox-1.0.7 | grep-2.4.2 | jug2.0 | mutt-1.4.2.1i | PurifyPlus.2003 | tidy-040106 | xdoclet-1.2.1 |
| cdrtools-2.00.3 | flex-2.5.4a | groff-1.19 | junit3.8.1 | mysql-4.0.20 | Python-2.4.2 | tiff-3.7.1 | xemacs-21.4.18 |
| cisam-7.26.UC1 | fontconfig-2.2. | gsl-1.5 | jython-21 | mysql-connector | QC_Coverage-3.0 | tiff-3.7.2 | xerces-2_6_2 |
| cscope-15.5 | fop-0.20.5 | gtk+-1.2.10fort | labelmerge-4.11 | nasaview-258s | qt-x11-free-3.3 | tix8.2.0b3up1 | xerces-C2_5_0 |
| cspice-60 | forte_for_java- | gtk+-2.4.0 | ldapjdk_4.16 | ncftp-3.1.8 | rcs-5.7 | tk8.4.6_threads | xerces-C2_6_0 |
| ctags-5.5.4 | freeglut-2.2.0 | gtkmm-2.4.2 | less-382 | ncurses-5.4 | readline-4.2a | tkcvs-7.1.1 | xerces-C2_7_0_s |
| cup-0.10k | freetype-2.1.5r | gtkspell-2.0.6 | libart_lgpl-2.3 | ndiff-2.00 | render-0.8 | tkdiff_4_0 | xft-2.1.2 |
| cvs-1.11.20 | fspice-60 | guile-1.6.4 | libgeotiff-1.2. | nedit-5.4 | Rhapsody | tkman-2.2rev1 | xpm-3.4k |
| db-4.2.52.NC | gawk-3.1.5 | gzip-1.3.3 | libghttp-1.0.9 | netbeans4.1 | rman-3.1 | tmake-1.8 | xrender-0.8.3r1 |
| ddd-3.3.8 | gcc-2.95.3rev2 | hpnp_d621 | libglade-2.3.6 | netpbm-10.18.12 | rsa-5.3.2 | top-3.5.1 | xrunner-6.0 |
| dejagnu-1.4.4 | gcc-3.4.3 | html2ps-1.0b3 | libiconv-1.9.2 | nitf-7.5 | rsync-2.6.3 | UconX-1.0 | xv-3.10arev3 |
| depot-5.13 | gcc-3.4.3_lib | hxplay-1.0.2 | libIDL-0.8.3 | ObjectSpace-2.0 | ruby-1.8.4 | UconX-1.1 | xxdiff-3.0.4 |
| derby-10.1.2.1 | gd-2.0.33 | ImageMagick-6.2 | libpng-1.2.5r2 | ObjMan-062900 | Sablot-1.0.1 | UconX-3.21 | zlib-1.2.3 |
| dia-0.92 | gdb-6.1 | indent-2.2.9 | libsigc++-2.0.3 | octave-2.1.69 | screen-4.0.2 | UconX-6.2 | zsh-4.2.0 |
| diffutils-2.8.1 | gdbm-1.8.3 | InstallAnywhere | libtool-1.5.20 | octave-forge-20 | sed-4.1.4 | unzip-5.50 | |
| doors_7.1 | gdbm_primitive- | ipf-4.1.3 | libunicode-0.4 | office52 | slang-1.4.9 | vim64 | |

JPL

# Google, Wikipedia, etc.

- OK, these are less about FOSS than about open and collaborative information exchange and problem solving.
- Does anyone look up mysterious error messages other places anymore?
- I wrote a glossary for a NASA white paper on a project I was new to; most of the entries came from Wikipedia :-)

JPL

# Sharing Lessons Learned

- http://llis.nasa.gov/ , Select “Topics”
  - Computers, Software, and lots of other non-software stuff that’s just neat to read
  - There are some pretty far-out gotchas written up
  - E.g. #1395: “...The design and code generation for the control and limit shutdown algorithms was relatively straight forward, but the challenge was to ensure it would all work correctly prior to firing up a developmental rocket engine for the first time...”

# My Challenge To You

- Do as much of my job as possible, so I can do something else!
  - The more of your code we send to Mars, the better
- Think ahead of the cutting edge
- Round out the total package
  - Automated regression test suites with the code
  - Documentation including Reference Guide, User Guide, maybe a QuickStart
- Brag about your users. Even in NASA, we don't like to be the first, only, or biggest users
- Your strength is in the people, the community. Conferences like this one help more than the developers

- THANK YOU for your hard work!