Open Source Software and its Role in Space Exploration

AFS/Kerberos Best Practices Workshop

June 14, 2006

djbyrne @ jpl.nasa.gov

Jet Propulsion Laboratory

California Institute of Technology

Common Goals

- FOSS (Free/Open Source Software) developers and NASA have a lot in common
  - Dedicated to expanding the pool of information floating freely through society
    - 1958 NASA Charter ”...for the full and open dissemination of the conduct of human spaceflight."
  - Focused on the cutting edge, creating tools and capabilities which did not previously exist
  - I like to think that the FOSS community's donations of code are reciprocated with knowledge about weather systems, climate, and basic science

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

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

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) and IP (Intellectual Property) are non-problems for existing Open code
    - ‘Though Adaptations and changes for mission details can be controlled and limited

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.

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

And now some examples

- CLARAty
- Electra
  - MRO
  - MSL
  - MSL TDS
- CCA
- Others

CLARAty Project

- Coupled Layer Architecture for Robotic Autonomy
  - http://claraty.jpl.nasa.gov/
- Investigating robot visions, navigation, operator interfaces, simulation challenges, etc.

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

CLARAty examples

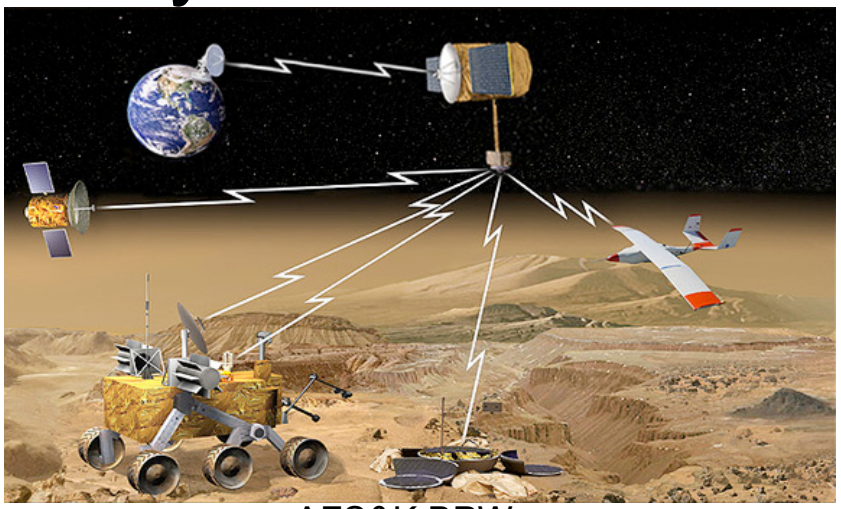
- Rovers, including next-generation for Mars, are being tested between many institutions
  - JPL, ARC, CMU, U-Minnesota, etc.
- Test images are shared via AFS
  - Latest pix from Spirit & Opportunity are used; AFS of course keeps the test set the same for everyone
- Code in AFS, CVS
  - YaM (Rapid Software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

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)

Electra Radios

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



Electra, 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/

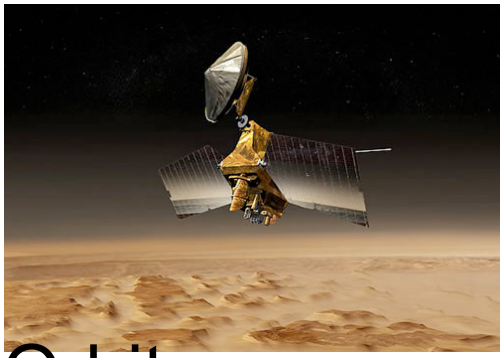
Electra, cont

- 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

Electra Development Environment

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

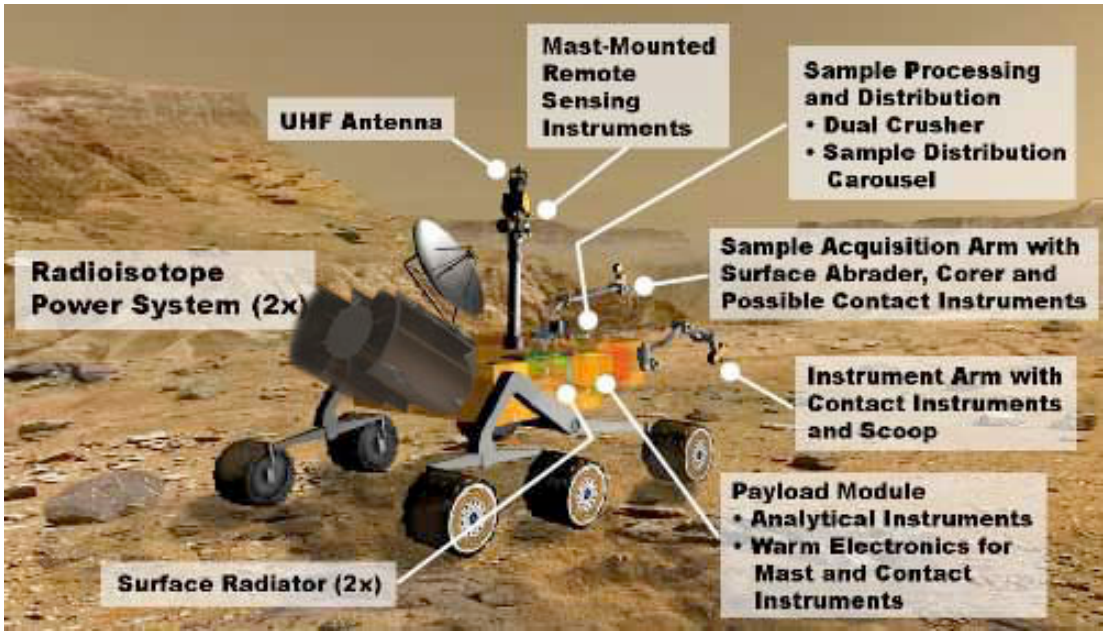
Electra: MRO



- 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



MSL Electra-Lite

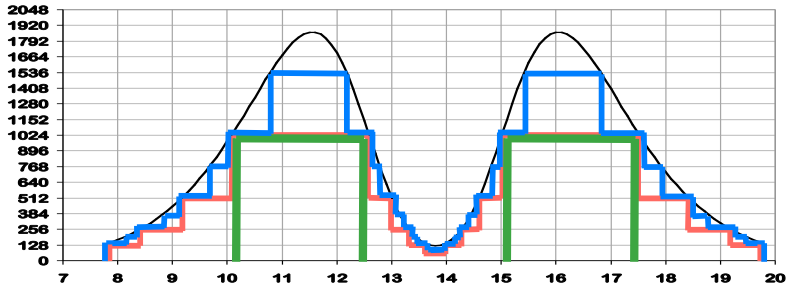
- Tighter mass/power constraints than MRO due to landing weight
- Data throughput requirements drive new software function in radio to adapt data rates during a communications pass by an orbiter

MSL Electra-Lite, cont

- Easiest sequencing for either commanding or data return is single data rate
- 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



MSL Electra TDS

- Terminal Descent Sensor
- Landing RADAR
- [111 seconds] video: EDL (Entry, Descent, and Landing)

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?

Lots of Other Examples

- Beowulf clusters all over the place
  - Aeronautical simulations
  - X-33 “Venture Star” updated space shuttle
- MySQL
- Mailman
- ...And on and on

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	

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
- Package automated regression test suites with the code
- 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. Workshops like this one help more than the developers

- THANK YOU for your hard work!