

# The Remote Agent Experiment

Debugging code  
from 60 million miles away

Ron Garret  
14 February 2012

# Themes

- Autonomous control of unmanned spacecraft
- Software development methodology and the limits of static analysis
- Effecting change in large organizations
- (Lisp)

[T]here is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things...

Niccolo Machiavelli  
“The Prince”, Chapter 6

[T]here is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things, **for the innovator has for enemies all those who have done well under the old conditions, and only lukewarm defenders in those who may do well under the new.**

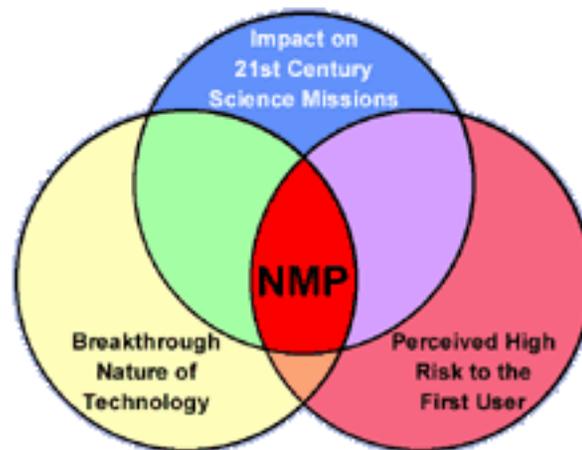
Niccolo Machiavelli  
“The Prince”, Chapter 6

# 1994: The era of big spacecraft is over

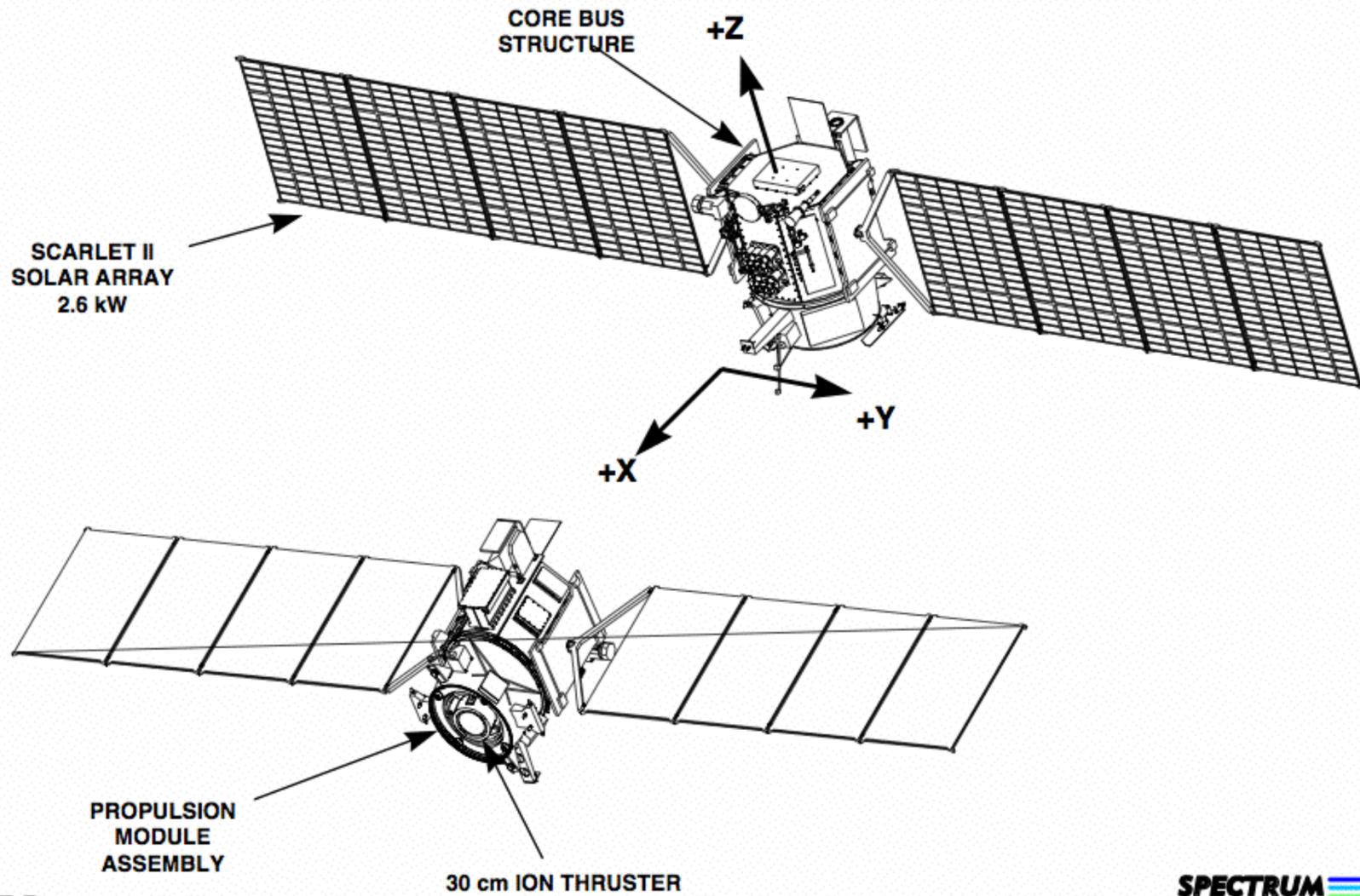


# New Millennium Program

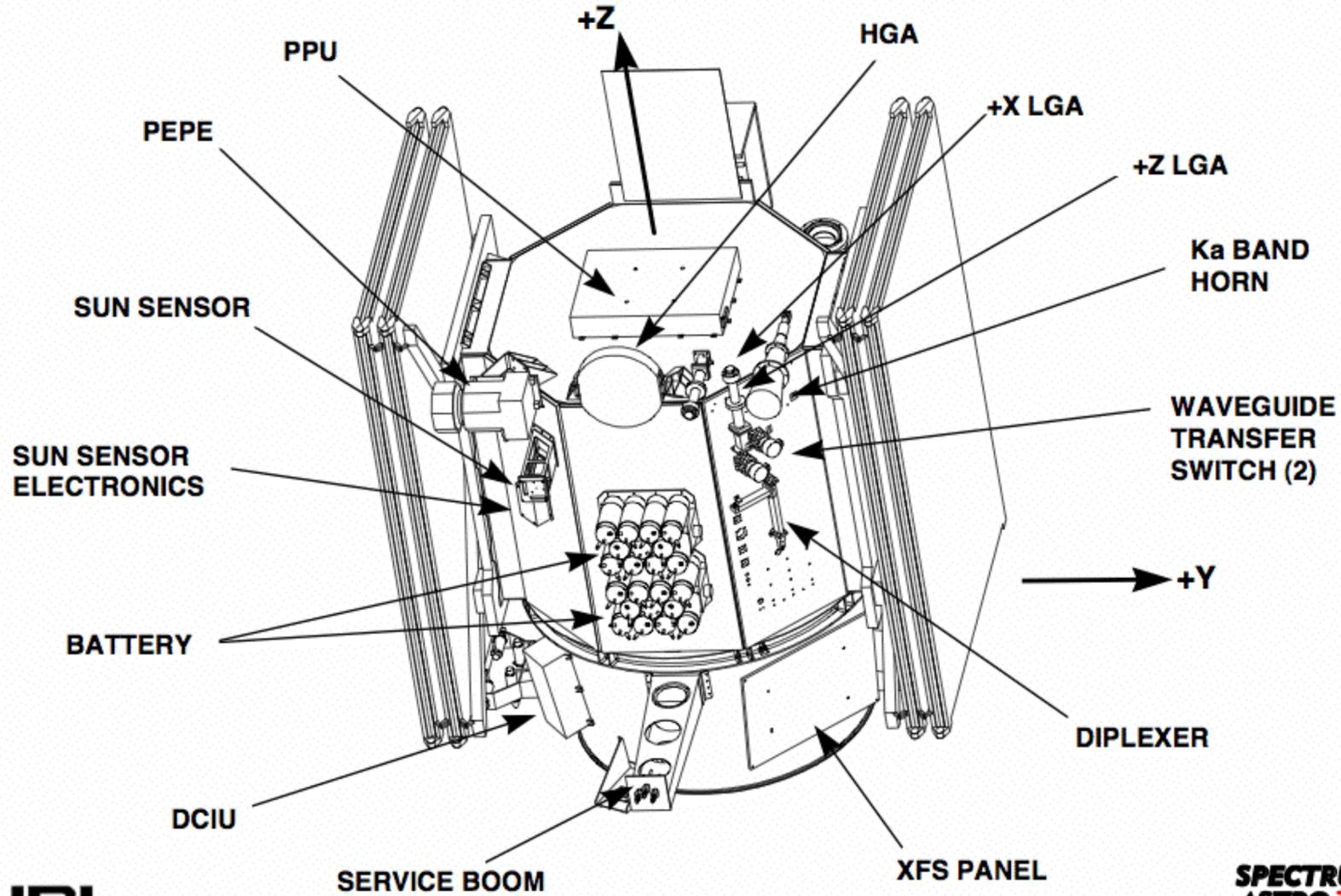
- Initiated by then-NASA-administrator Dan Goldin in 1994
- New technology, economies of scale...
- “Better, faster cheaper...”

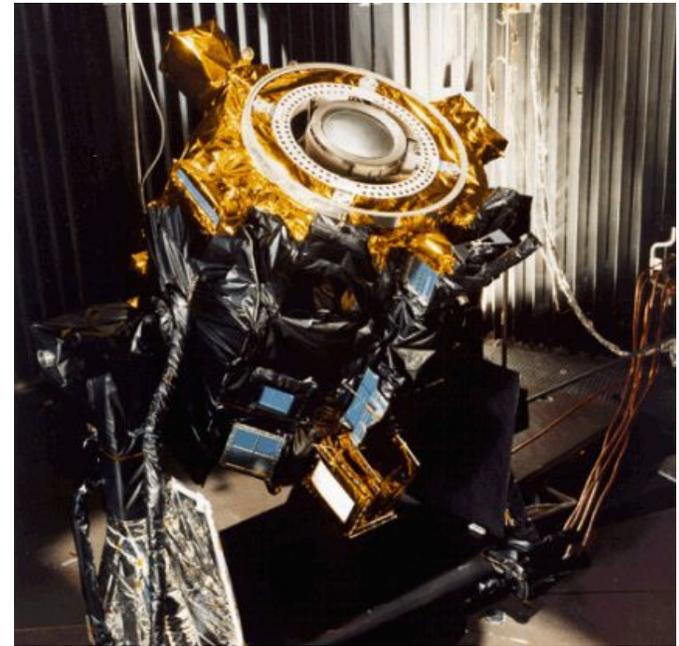
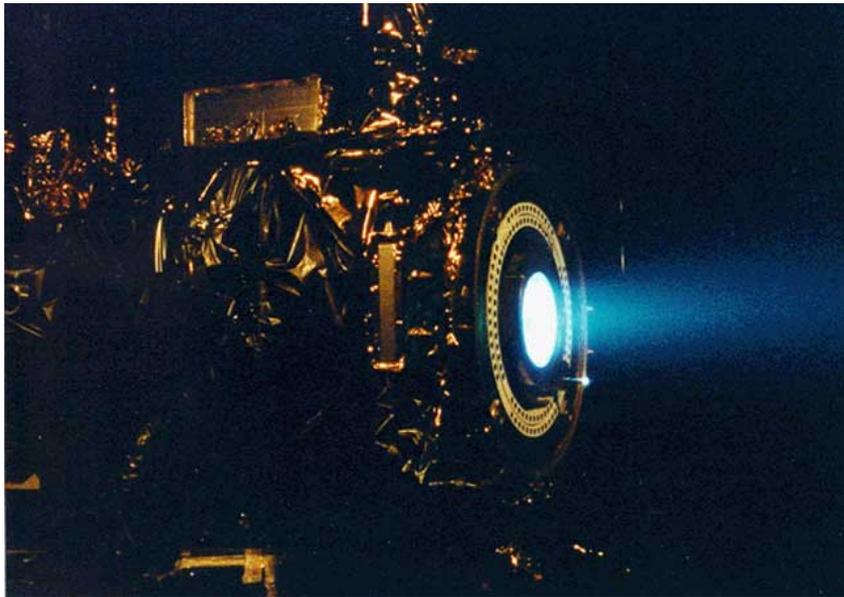


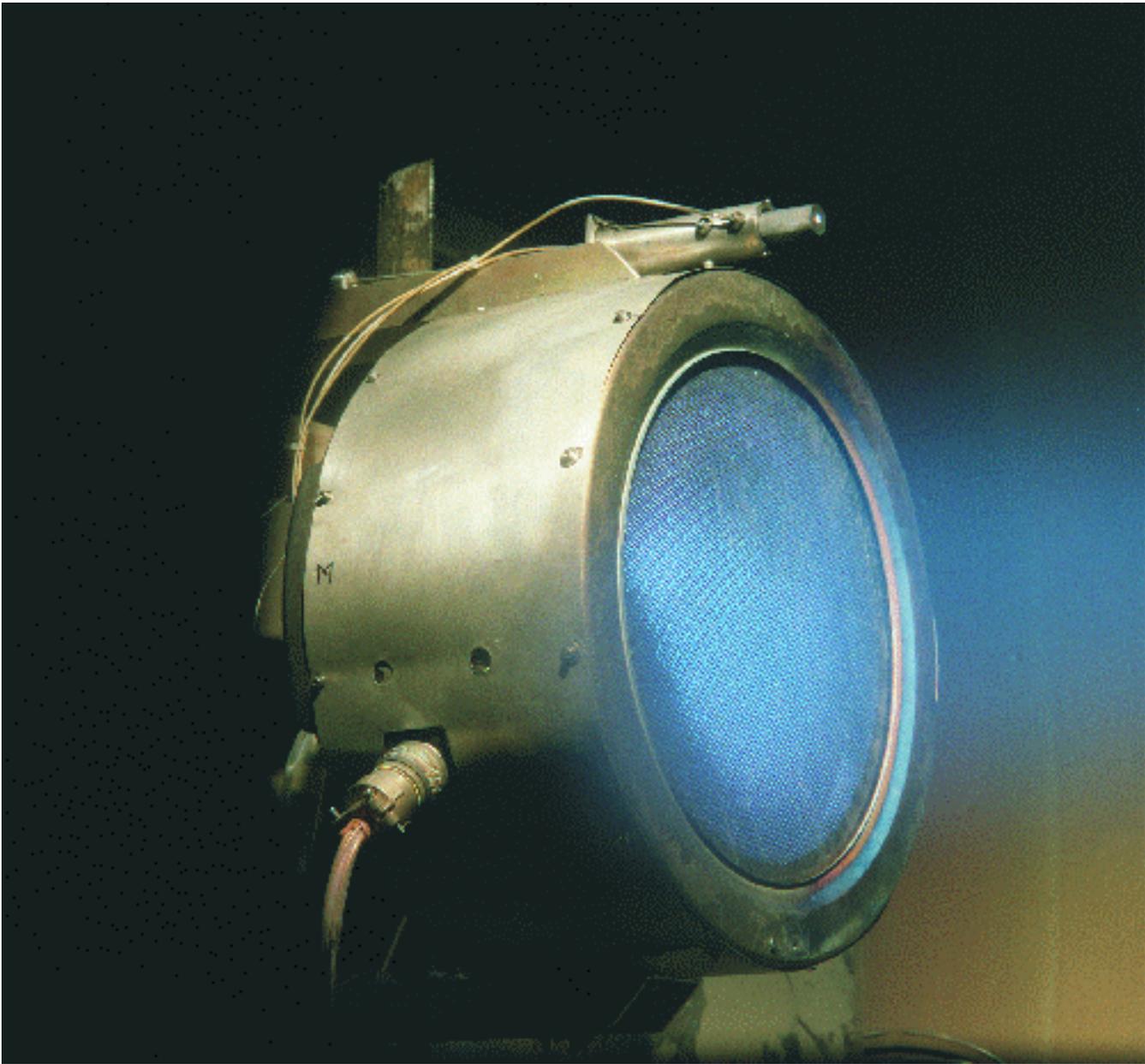
# NMP DS1 SPACECRAFT BUS GENERAL CONFIGURATION

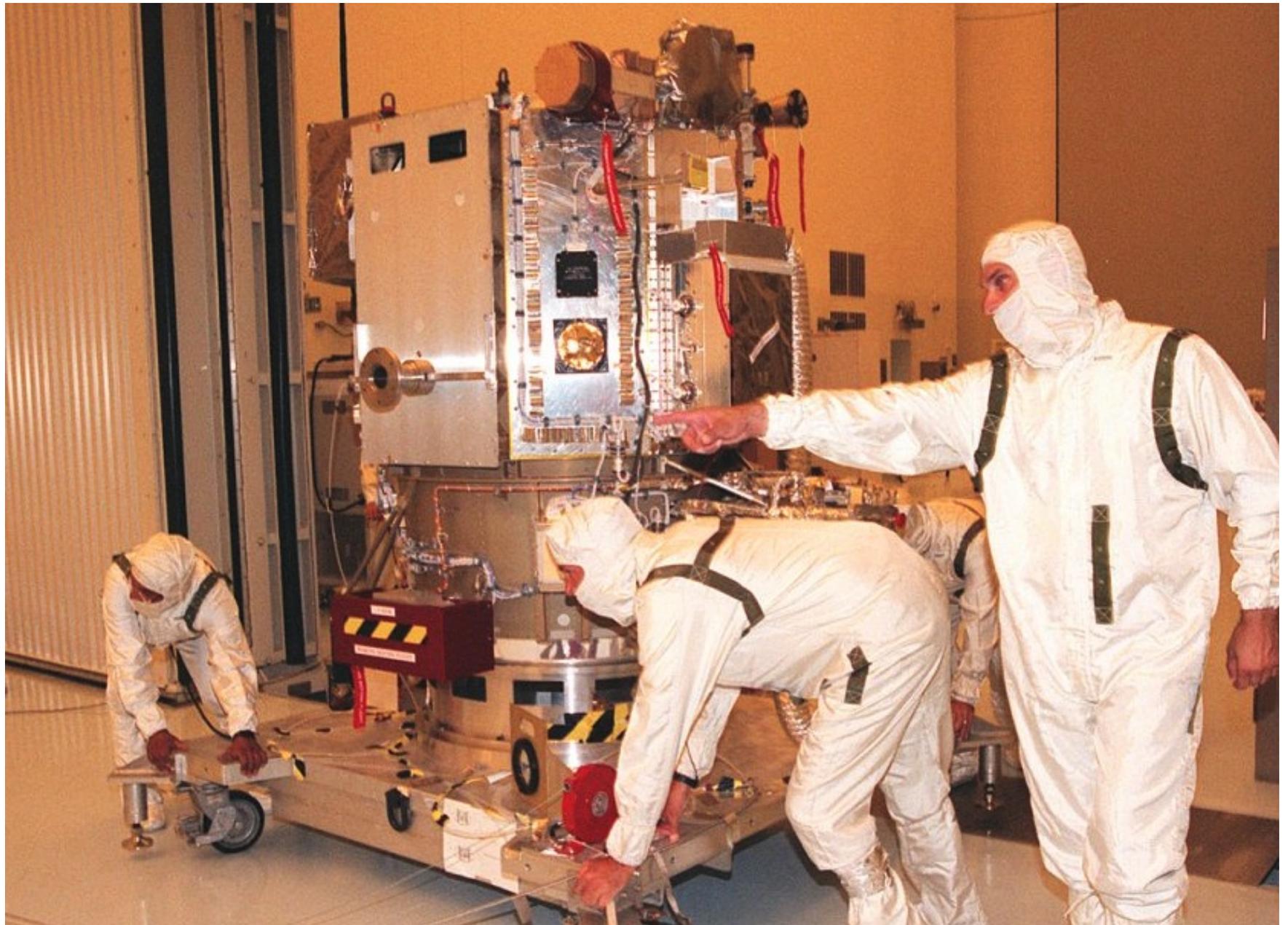


# EQUIPMENT ARRANGEMENT









# DS1 vs Galileo and Cassini

- Galileo (1977-1986/9)
  - ~10 years inception to launch
  - ~2500 kg
  - \$1.7B
- Cassini (1982-1997)
  - 15 years inception to launch
  - ~2500 kg
  - \$3.2B

# DS1 vs Galileo and Cassini

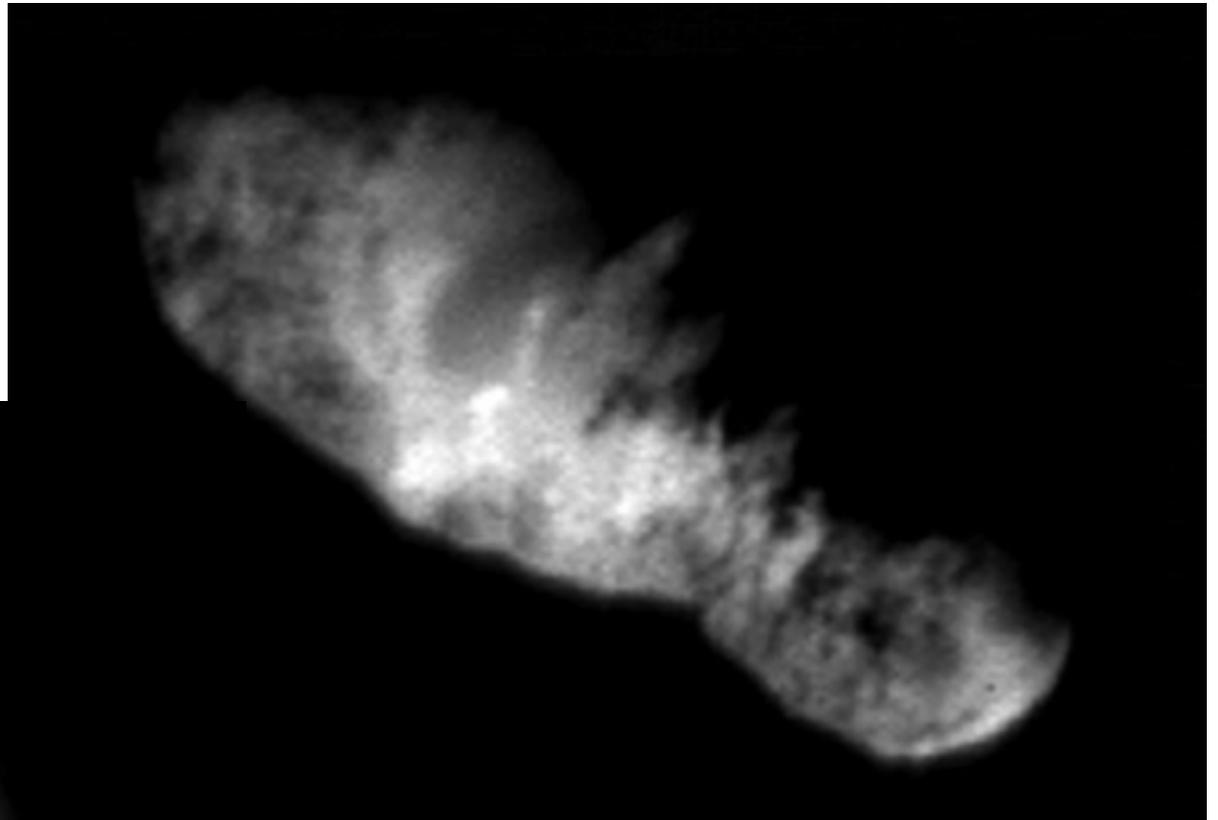
- DS1 (1994-1998)
  - 4 years inception to launch (25% of Cassini)
  - 373 kg (15% of Cassini)
  - Originally budgeted at \$100M (3% of Cassini)
  - Final mission cost: \$150M

# DS1 Technology Demonstrations

- Solar-electric ion propulsion
- SCARLET concentrating solar array
- Auto-nav visual navigation system
- Small Deep Space Transponder (SDST)
- Beacon monitor, PEPE, a few others...
- And, of course, Remote Agent

# DS1 Science

Asteroid Braille

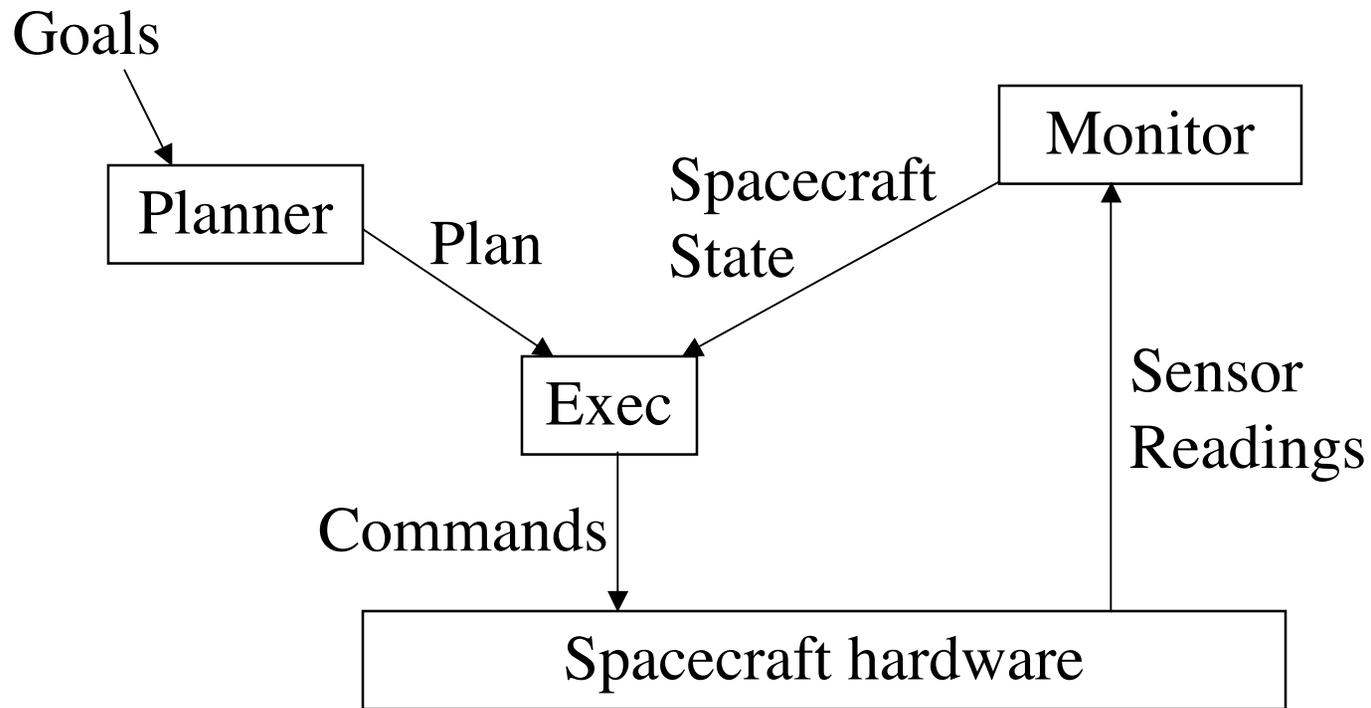


Comet Borrelly

# Remote Agent

- Spacecraft control without human supervision
- Intended to reduce operations costs
- Three major components:
  - Planner (Europa, ARC)
  - Executive (EXEC, JPL)
  - Diagnostic system (Livingstone, ARC)

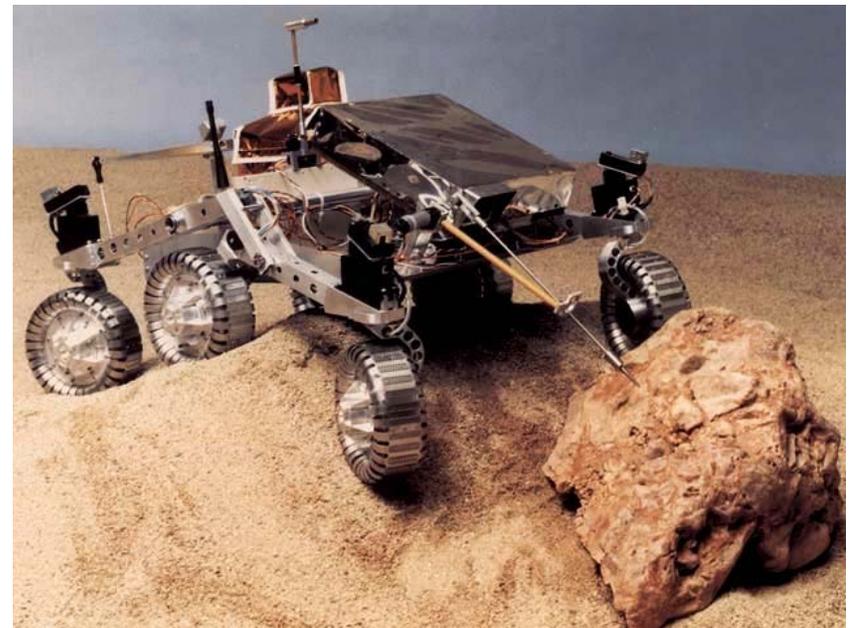
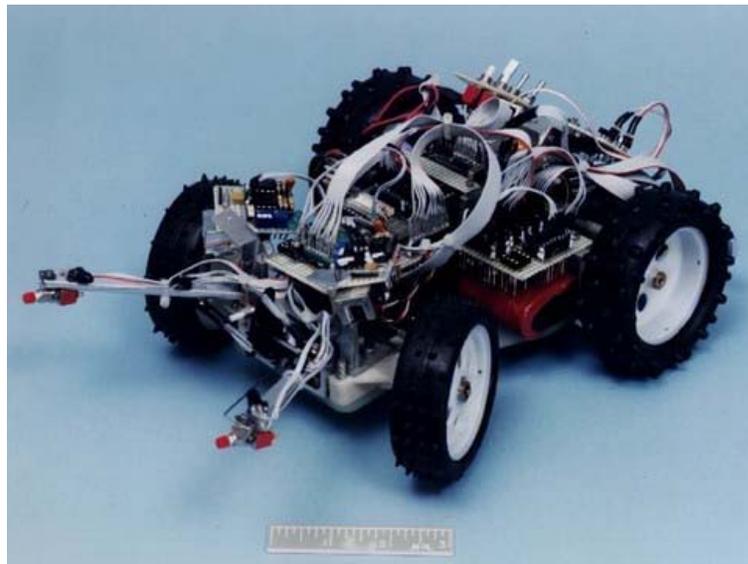
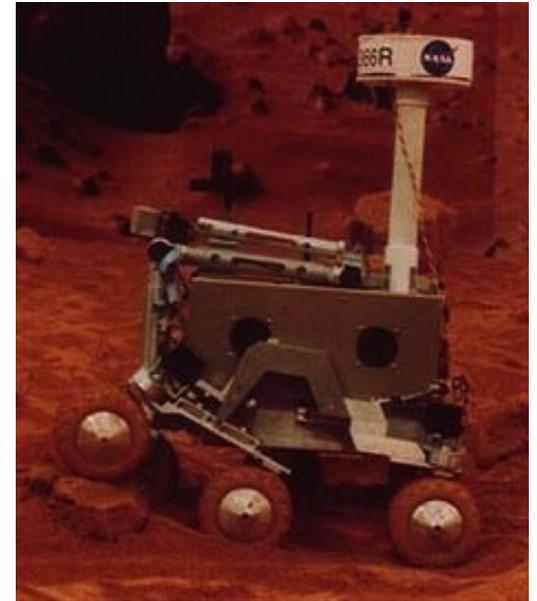
# Remote Agent



# “Lisp” is a four-letter word

- Remote Agent was written (mostly) in Lisp
- This was the mid-90's
  - Galileo flight software written in HAL/S
  - Cassini flight software written in Ada
  - Python 1.0 had just been released
  - Java was still a year in the future
  - Ansi C was “new” by S/C standards (only 5 years old)
  - C++ was (still) not ready for prime time...

# Rovers



# SHARP: Spacecraft Health Automated Reasoning Prototype

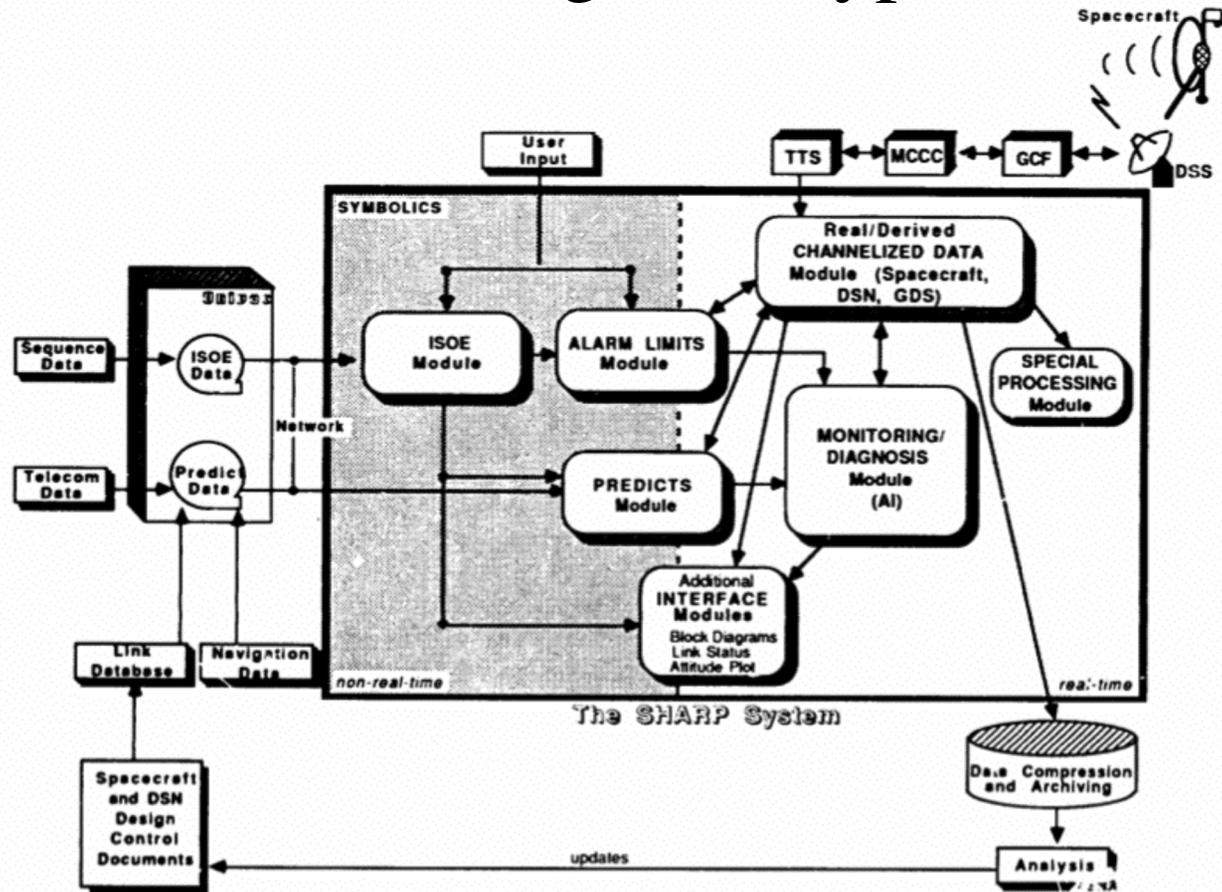


Figure 1. SHARP Telecom System Overview

# Other Lisp Projects

- Plan-IT2, a spacecraft sequence generator
- Galileo magnetometer patch
- MeSMER (Modular Simulator of Mobile Robots)
- TSim, a modular spacecraft simulator
- SDS (State Database System)

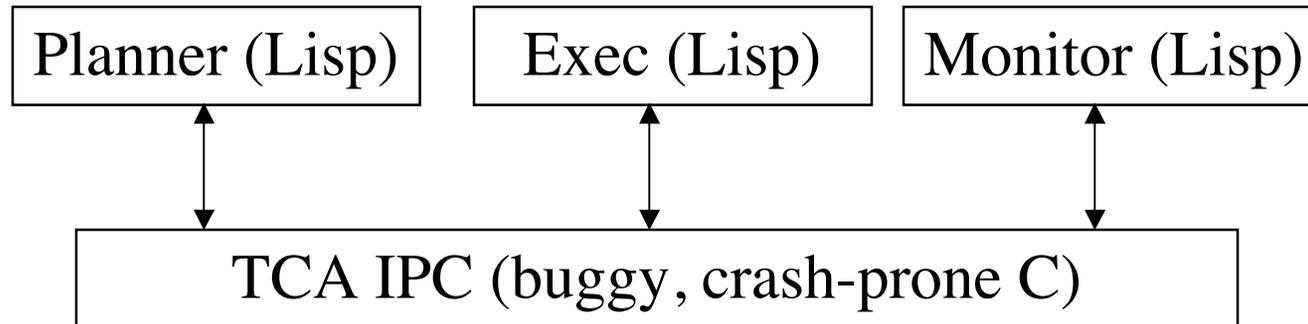
# RA's fourth component: TCA

- TCA: Task Control Architecture (Simmons)
- Early message-passing architecture
- Developed at CMU
- Two major components
  - TDL (Task Description Language)
  - IPC (Interprocess Communication)

# Design by Politics

- IPC competed with CORBA
- TDL competed with ESL (Exec)
- Political considerations dictated that CMU get a piece of the action
- CORBA didn't have an advocate

# The unfortunate result



# It gets worse

- Data has to be marshaled and unmarshaled from Lisp to C and back again
- TCA IPC has its own data description language
- DDL must be manually synced with .h files and Lisp FFI
- CLASH (Kanefsky)

# Disaster looms

- RA was viewed with extreme suspicion from the beginning
- TCA IPC crashed regularly
- Entire DS1 project is over budget, behind schedule
- Critical design review:
  - NASA big wig to lead software integration engineer: “What is the one thing you would change to make things better?”
  - Lead SWIE replies: Dump Remote Agent.

# The Decline and Fall of Western Civilization

- RA downgraded from mainline flight software to a flight experiment (RAX)
- Attempt to rewrite planner in C++ failed

# Chronology

- 1994 - NMP announced
- 1996 (approx) - RA downgraded to RAX
- Oct 24, 1998 - DS1 launched
  - RAX software not complete
- Feb 1999 - RAX software uploaded
- May 1999 - Remote Agent Experiment(s)
- July 28, 1999 - Braille flyby
- Sept 22, 2001 - Borrelly flyby

# DS1 “issues”

- Ion engines shut down 4 minutes after initial start
- Software crash 48 hours prior to Braille flyby (unrelated to RAX)
- Target tracking failed at Braille
- Star tracker failed shortly after Braille
- The RAX bug...

# The RAX bug

- Reliability was *extremely* important
  - We were going to control a \$150M asset
  - Couldn't go to the machine room and reboot in the event of failure
- RAX was *extensively* tested for *months*
- Exec “proven” correct using SPIN

# The RAX bug

- RAX ran for three days, heavily scripted scenario
- On day 2 an expected event failed to occur
- Interesting debugging process
  - \$150M asset, 45 light-minutes away
  - Having a REPL was invaluable
- Problem turned out to be a race-condition leading to deadlock
  - “Proven” impossible
  - Never happened in ground testing

# The RAX legacy

- Officially RAX was a success
  - It (mostly) worked
  - We didn't lose the spacecraft
  - Flying a REPL was revolutionary, RAX bug could not have been fixed without it
- Unofficially it was a disaster
  - No spacecraft has flown autonomously since RAX
  - No NASA software development has been done in Lisp since RAX



# What went wrong?

- It was not just RAX that failed...
- New Millennium was supposed to save NASA. It didn't.
- Deep systemic failure of NASA's attempt to reinvent itself. Why?

# What went wrong?

- We did not lack support from upper management
- We did not lack technical expertise
- We did lack spacecraft training, but we learned on the job

# What went wrong?

- What we lacked was *a true understanding of our user's needs*
  - Focused on technology, economics, coolness
  - We assumed the user wanted tools to make them more productive.
  - What they really wanted was *job security*

# NASA: RAX writ large

- NASA is *not* about space exploration. It has *never* been about space exploration.
- NASA *was* about beating the Soviets
- Since 1991 it has been about ... ???

The hardest part of giving  
the customer what they want  
is figuring out what it is.

Backup slides

# What do your users really want?

- Users are not necessarily customers
  - Your customers are the people who give you money
  - Google's users are *not* Google's customers
- In a large organization, *your* users are often your *colleagues*
- Intra-organizational economics are different
  - Good will, favors, and making people look good to their superiors are the local currency, not money

# Advice

- Innovation is *always* disruptive
  - Find out whose toes you are stepping on (not always obvious)
  - Shmooze them, or get them out of the way, but *do not ignore them*

# More RAX legacy

- VxMCL was not the flight Lisp. We used a custom port of Harlequin.
- CLisp was also a candidate, successfully ported to VxWorks, but ultimately rejected because of lack of multithreading

# More RAX legacy

- RAX is the reason Clozure Common Lisp exists
  - Gary Byers hired to port MCL to VxWorks
  - Along the way ported to Sun Sparc
  - That code base eventually became OpenMCL, which is now CCL