Maintaining Ground Software and Hardware Over Missions That Span Multiple Decades

Lessons from the New Horizons Mission

Daniel Hals
New Horizons Ground Systems Lead
Johns Hopkins Applied Physics Laboratory (APL)
daniel.hals@jhuapl.edu
(240) 485-6385

Agenda

- New Horizons Introduction
- The Challenge – Ground System Freezing
- Why Do Freezes Have a Larger Impact?
- Factors to Consider from NH Lessons Learned
- Summary

New Horizons Introduction

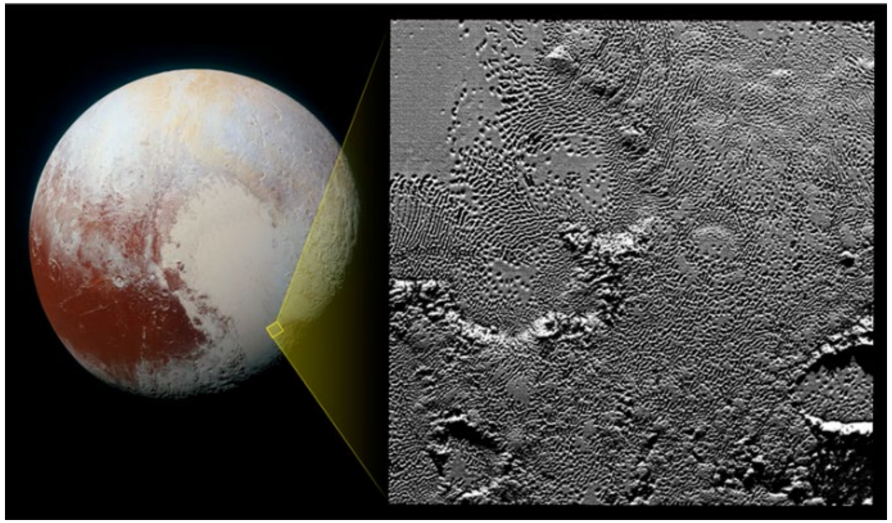
- The New Horizons (NH) spacecraft was the first mission to Pluto, completing the space-age reconnaissance of the planets that started 50 years earlier
  - It provided the first images of Pluto
  - It was also the first mission to explore the solar system's recently-discovered "third zone," the region beyond the giant planets called the Kuiper Belt.

- Mission timeline:
  - 2002 – APL development starts
  - 2005 – APL Integration and Test starts
  - 2006 – Launched on January 19
  - 2015 – Extremely successful flyby of Pluto on July 14
  - 2019 – Extremely successful flyby of Arrokoth on January 01
  - 2050 – Continued operations

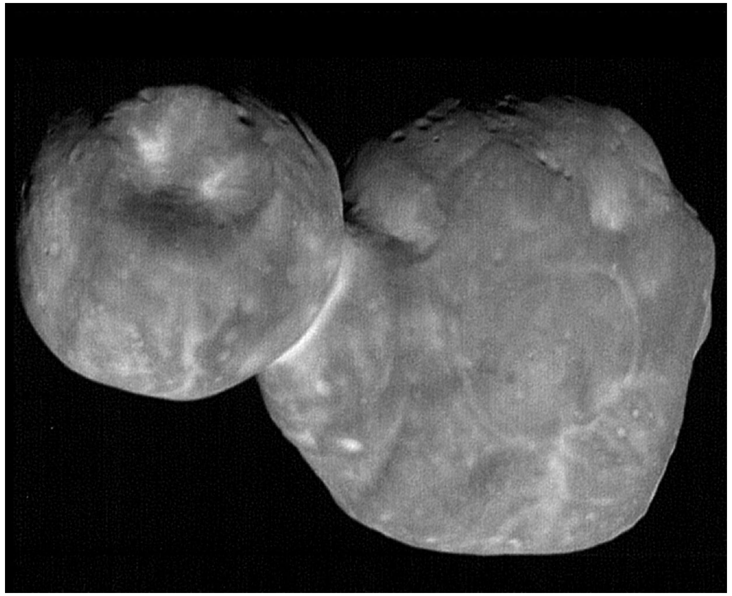
- Operations in the 2030s and beyond assumes an adjustment in the baseline power draw

- http://pluto.jhuapl.edu

Pluto: July 14, 2015



Arrokoth: January 01, 2019



The Challenge - Ground System Freezing

- Missions typically freeze the Ground System on a known good baseline prior to significant activities
  - For New Horizons, this occurred ~6 months before both the Pluto and Arrokoth flybys

- There are many valid reasons for a Ground System freeze:
  - Assures the tested versions match the operational versions for important activities
  - Limits the possibility of new changes breaking existing functionality
  - Reduces the number of changing variables when problems are encountered
  - Supports legacy hardware/software that cannot be easily updated
  - Conveys important activities to all members that support the Ground System, especially those who support multiple missions

- The challenge is when Ground System freezes are never lifted
  - “If it ain’t broken, don’t fix it” can be a very attractive policy to management

- A freeze adds “technical debt” the longer it is in place
  - Technical debt increases the complexity of making even trivial changes

Why Do Freezes Have a Larger Impact?

- A favorite question asked when discussing this topic is “what do older NASA missions do, such as Voyager?”

- There are three significant evolutions with technologies in a Ground System:
  - Hardware with a shorter life expectancy
    - In general, technology is moving towards cheaper solutions that expect to be replaced more frequently
  - Requirements that are continually updated or require continual updates
    - Information Assurance (IA) security is a great example
      - Vulnerabilities continue to be discovered, such as the log4j Critical vulnerability in 2021
  - Ground Systems are becoming more distributed
    - How many team members reside in locations off-site from the Ground System hardware?
      - How does the data get routed between the antenna and the Ground System?
      - Working from home as a result of COVID
        - Certainly was not part of the plan in Phases A-D
    - The Ground System then becomes dependent on the requirements / updates for each external interface

Factors to Consider from NH Lessons Learned

- The expected lifetime of a mission
  - If the spacecraft will run out of a critical resource in ~5 years, the risk of a freeze may be acceptable
    - For example, purchase a lifetime supply of hardware spares

- External interfaces should be expected to change, such as:
  - Interfaces with the Deep Space Network (DSN)
  - Interfaces with the Science Operations Center (SOC)
  - Protocol changes, such as FTP to SFTP

- Public-facing interfaces should be expected to change even more frequently, such as:
  - Websites
    - SOAP vs. REST
    - Version of HTML
    - Changes in Web browser support
  - Shared data centers, such as cloud-based storage

Factors to Consider from NH Lessons Learned

- End of Life plans for software languages used
  - Are major updates expected? Will new features in new versions be desirable to incorporate?
  - Will the language itself continue to exist?
    - Perl vs. Python
    - Will Matlab exist in the 2050s?
  - How will the development environment change?
    - IDEs
    - Compilers
    - Software Languages
    - Repository languages
  - Will new engineers be trained in these tools?
    - Do they teach Fortran in college?
    - Will they teach C++ in the 2050s?
    - Do they teach SVN in college? Will they teach GIT in the 2050s?

Factors to Consider from NH Lessons Learned

- End of Life for Hardware architecture and Operating Systems
  - SPARC vs. x86
  - Older operating systems cannot be installed on newer hardware
    - Windows XP, Solaris 6, etc
  - How long does the vendor support the product?
  - Is virtualization available? Does it meet performance needs?

- Security requirements
  - Do Operating Systems need to be patched?
  - Will older versions be allowed on the required networks?
    - Both internal networks and external networks, such as the NASA Restricted IONET
  - How do critical vulnerabilities get addressed?
    - Will these provided by a vendor?
    - Are these expected to be provided by the Ground System team?

- The need for a mission to react to unpredictable changes – technology changes quickly
  - Will Java even exist in 2050 or will it be viewed like Fortran?

Summary

- Many questions have been raised but there is no “right answer”
  - Each mission must assess their requirements, baseline technologies, risk posture, predicted lifespan
  - The point is to ask the question and revisit the decision as needed
    - Which is preferable: updating hardware or performing a lifetime purchase of spares?

- Imagine having to define the following, which will be used in 50+ years:
  - Integrated Development Environment (IDE)
  - Bug tracking system
  - Hardware architecture platform
  - Third party tools
  - External Interface Control Documents (ICDs)
  - Security posture

- How can future engineers, who have not even finished high school, or in some cases aren’t even born yet, be responsible for maintaining this baseline?

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY