Tails of Developing a Common Ground Architecture

Presented at: GSAW2006

L. Herrera
JHU/APL Space Department

APL

The Johns Hopkins University
APPLIED PHYSICS LABORATORY

Content Overview

- **Background on JHU APL ground systems**
- **The Common Ground Architecture Approach**
- **Benefits**
- **Lessons Learned**

APL

Vision of Common Ground Architecture

- **Support multiple missions with a common architecture approach and code infrastructure**

  - **To develop with reuse in mind**

  - **Find the common requirements across missions to build to**

- **Encapsulate mission specific details within a hierarchy of classes**

  - **Allow common areas of code to interface with a base level abstraction representation**

  - **Appropriate mission definitions are addressed in the implementation of an abstract class**

APL

Heritage: Single Mission Development

- **A Dedicated Teams was formed for each mission.**
    - **+ No conflicts in resource scheduling across programs**
    - **+ Responsive to the specific needs of the mission**
    - **- Knowledge transfer & sharing between teams not inherently facilitated**
    - **- Potential for redundant tasking**

- **A Snapshot of a previous system was used as basis for next mission**
    - **No requirement to design for reuse**
    - **+ A new mission did not have to start from scratch**
    - **+ Users were familiar with overall functionality of applications**
    - **- Not a simple or straight forward task**
    - **- Modifications for some areas could be comparable to complete rework**
    - **- Occasional need for complete rework or creation of new applications**
    - **- Fixes or enhancements not easily shared between active missions**

- **Approximately the same efforts was required to provide the same fundamental functionality for each mission!**
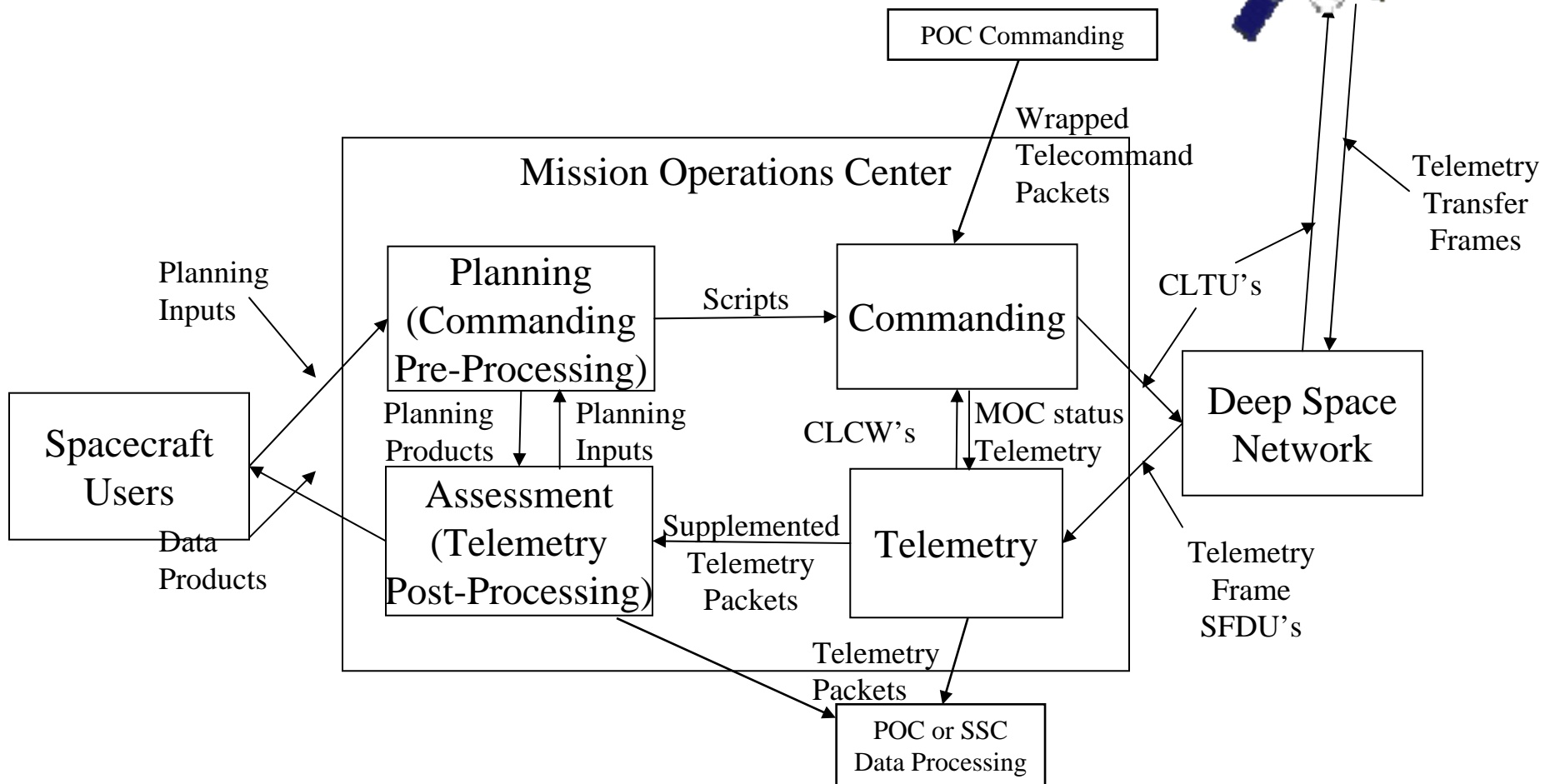
APL

Need for Change

- **JHU/APL was awarded with three NASA missions with overlapping schedules.**

- **Snapshot approach for "re-use" worked for missions with dovetailed schedules, where teams can easily transition between projects**

- **Needed a solution to address demanding schedules and work effort with limited resources**

- **Would like to:**
    - **Reduce overall development costs**
    - **Reduce required staff per mission**
    - **Improve quality**
    - **Improve estimations in scheduling and cost**
    - **Support DoD and NASA missions**

Common Ground Architecture Approach

- **Examined traits of heritage ground systems**

- **Staff reorganized into teams centered around "Product Lines."**
  - **5 Product Lines Created:**
    - **Assessment, Commanding, Planning, Telemetry, Tools***

- **Large effort to direct 3 missions to a common set of requirements for the ground system software**
  - **Still addressed mission specific requirements and sub-requirements**

- **Identified heritage software that could be converted to new approach**
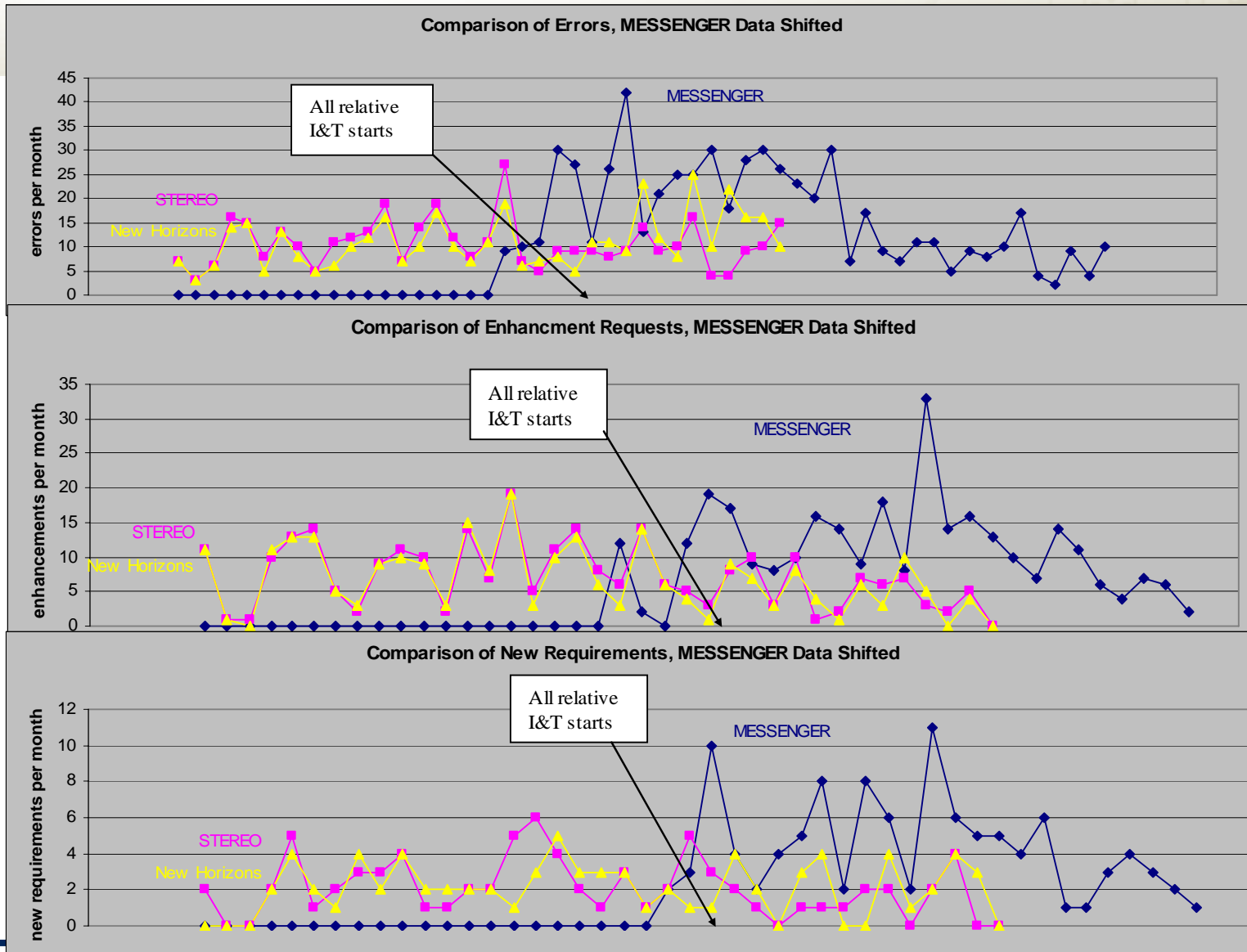
- **Required new CM support to handle code infrastructure**

APL

Common Ground Architecture Cartoon

POC Commanding

Wrapped Telecommand Packets

Mission Operations Center

Planning Inputs

Planning (Commanding Pre-Processing)

Scripts

Commanding

CLTU's

Telemetry Transfer Frames

Planning Products

Planning Inputs

CLCW's

MOC status Telemetry

Deep Space Network

Spacecraft Users

Data Products

Assessment (Telemetry Post-Processing)

Supplemented Telemetry Packets

Telemetry

Telemetry Frame SFDU's

Telemetry Packets

POC or SSC Data Processing

Common Ground Approach Benefits

- Shared cost of development across multiple NASA missions
- Reduced redundancy & capitalized on domain knowledge
- Supporting needs of all current missions
- Increased familiarity among users
- Decreased development time & cost for subsequent missions
    - Telemetry archiving system can be brought on-line for new mission in a week
- Solid base established
    - Free up resources to add new layers of functionality
- Increased reliability through repeated testing & use
    - Metrics show decrease in SW Change Requests for each subsequent mission

APL

Comparison of Change Requests
Shifted for to Normalize Development Phase

Comparison of Errors, MESSENGER Data Shifted

Comparison of Enhancment Requests, MESSENGER Data Shifted

Comparison of New Requirements, MESSENGER Data Shifted

Lessons Learned

APL

What Could Be Done Better Brainstorming

- Had a chance to catch our breaths and prepare for the next round of missions
- Held numerous brainstorming meetings to gather input from greater community of users
  - Ground & Flight software developers
  - Hardware developers
  - Software test and verification teams
  - Integration & Test conductors
  - Flight operators

Common Ground Approach Challenges

- **Increased cost of initial development**
- **Increased difficulty in coordinating resources across simultaneous program schedules**
- **Increased resources & staff over commitment on occasion**
- **Increased need to negotiate requirements among multiple missions**
- **Sophisticated configuration management system is needed**
  - **False assumption that all missions stay w/ agreed approach stay consistent with main line development**
  - **Highly complex due to missions freezing their versions and creating branches**
- **Tightly manage modifications to "generic" classes**
- **Complexity of system configuration remains an issue**
  - **Problems now reported due to configuration**
- **Not true "reusable" software**
  - **Several areas are mission specific and numerous update to "common" code**

High Level Areas to Improve from Users

- **Insight into system status and data flow from a single interface**
  - **Component & network health and location of data**
- **Seamless integration of data access across real-time and archived sources**
  - **Walk forward and backward as desired in data without having to query separate components**
- **Enhanced trending and plotting**
- **TIVO like capabilities for TT&C actions and telemetry**
- **Central system configuration**
- **Improved support for telemetry and command definitions/modifications for individual and team use**
- **Removal of dependency to any third party software**
  - **Don't allow any one component drive overall architecture**

APL

What To Tackle Next

- **Currently evaluating Common Ground Approach and functionality**
  - **Working to identify generic services and framework outside of current implementation to help new architecture designs**
  - **Incorporate User feedback**

- **Open to modify or complete rework of architecture if necessary**
    - **Support additional communication approaches**
    - **Decouple several "over grown" components**
    - **Plan for change in multiple areas**

- **Become less dependant on any one vendor solution**

- **Search for solutions & technologies offered by other organizations**
  - **Identify emerging technologies that JHU/APL could help cultivate**

- **Research cutting edge solutions in automation and integrated services and functionality not current operational use**
  - **Knowledge gained in research efforts fed back to operational development on recommendations to improvements**