MPCS: Develop and Test As You Fly for MSL

GSAW 2008

Michael Tankenson & Lloyd DeForrest

Wednesday April 2, 2008

Mission data Processing & Control System (MPCS)

- An Advanced Multimission Operations System (AMMOS) product line for Mission Control and Testbed functions, with strong Flight Software (FSW) Development & Testbed support (uplink, downlink functions)

- Support for all Mission Phases
  - Early FSW testing (on individual developer workstations) to
  - High Fidelity Testbeds (System Test Lab) to
  - Assembly, Test and Launch Operations (ATLO) to
  - Operations (fully distributed configuration with DSN interfaces)

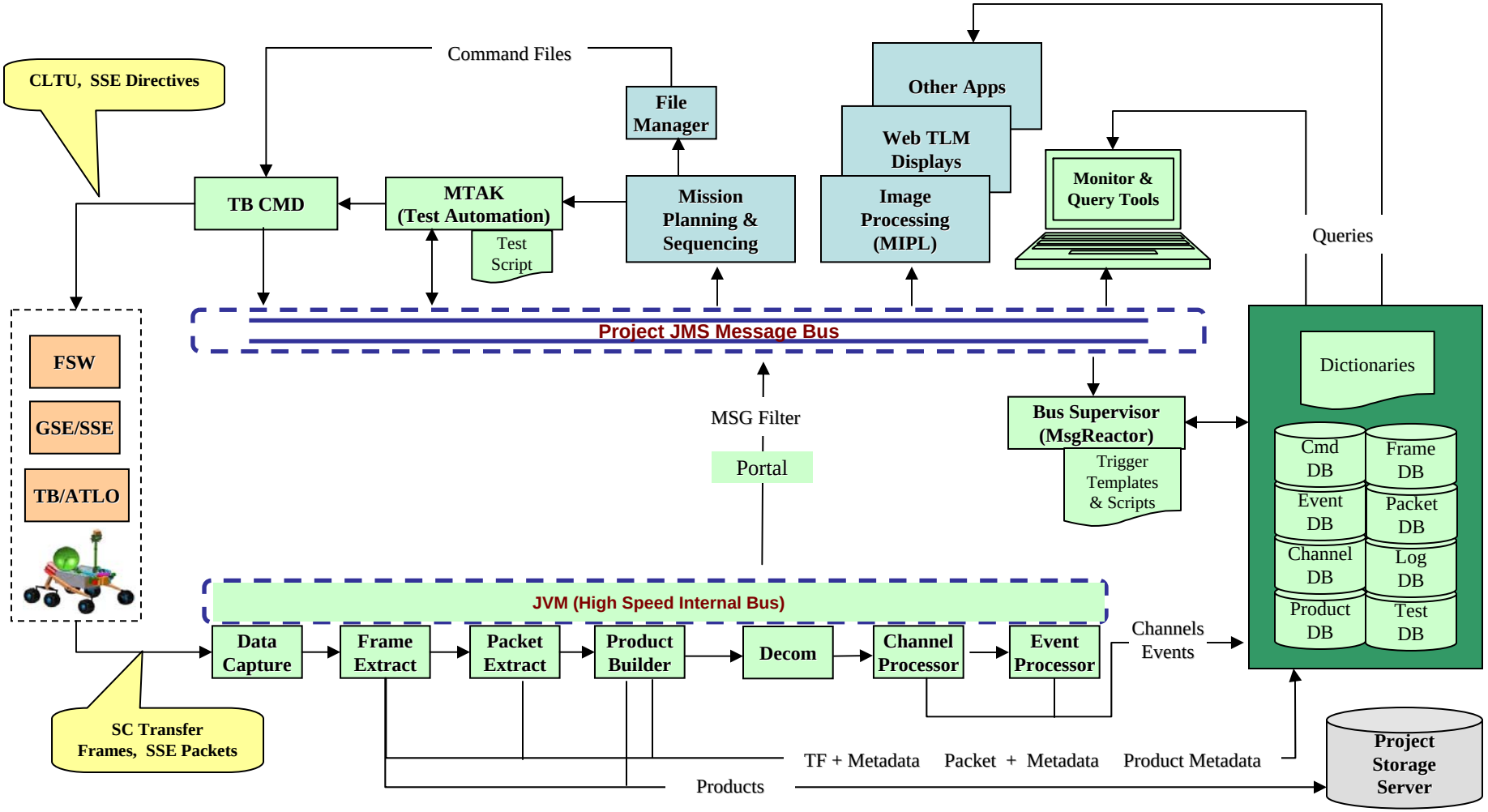
- MPCS components are Java-based, platform independent, and are designed to consume and produce XML-formatted data

- Inheritance from previous flight projects (e.g. Mars PathFinder, Mars Exploration Rovers, Mars Reconnaissance Orbiter, DAWN)

- MPCS component architecture based on an event driven model with Java Messaging System (JMS) messaging used to trigger events and status
  - Enables plug and play architecture
  - Industry standard reliable delivery of data and distributed system design

- MPCS is the first major application implemented based on the JPL DSN Information Systems Architecture (DISA)

MPCS, A Generic View



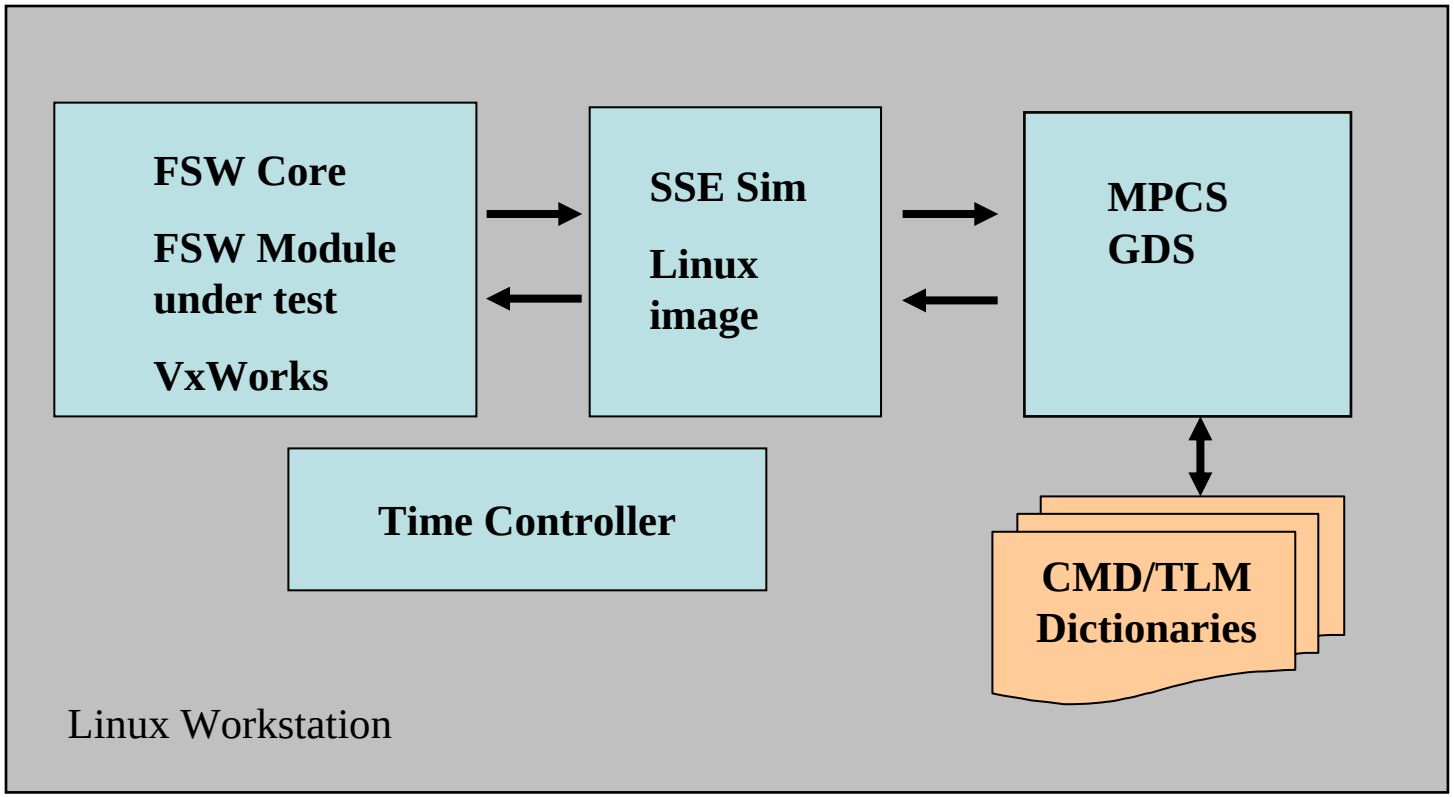
Test-As-You-Fly (TAYF) vs Develop & Test As you Fly

- NASA Lessons Learned (TAYF)
  - The principle of "test-as-you-fly" means that ground tests and simulations should accurately reflect the planned mission profile, plus margin and the appropriate off-design parameters
  - In other words, the pre-launch testbed and ATLO environments should closely match the Ground System (GS) that a mission will be flying with

- Develop & Test As You Fly*
  - Extend the same concept to FSW development environments
  - From the very first phase of a mission, when flight software is being developed, use the same GS that the mission will fly with
  - The GS is used as a test tool early on
    - For FSW development to verify proper uplink and downlink behavior and dictionary compatibility
    - In the Testbeds, used to capture all data associated with test sessions for later replay and analysis

*** Courtesy of Joe Kahr, MSL GDS System Engineer**

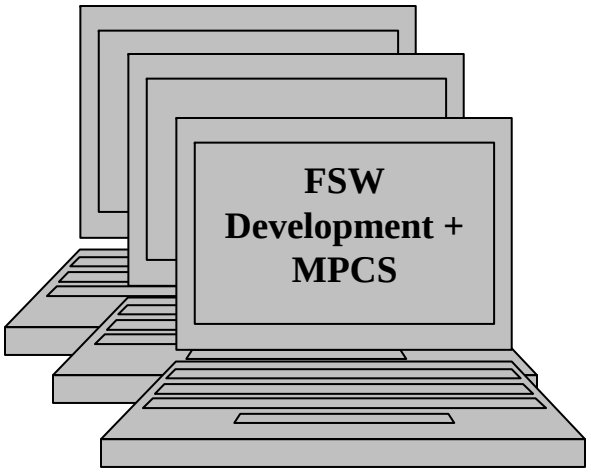
Typical MSL Workstation TestSet



FSW (VxWorks) image
- SSE Simulation (Linux) image
- MPCS GDS support infrastructure (Linux)

Both images (FSW, SSE) run as Linux processes with a synchronizing task under the Python Test Framework (PTF)
Communication is through Shared Memory
Communication to MPCS is though socket to local CHILL task for command and telemetry functions
CMD and TLM defined by XML dictionaries corresponding to FSW version generated by FSW and available to MPCS

Advantages and Disadvantages of Develop-As-You-Fly



- Advantages
  - Earliest use of flight-like Ground System
  - Everything contained on a single workstation
  - Multiple FSW Developers working independently, each with their own version of CMD/TLM/Product Dictionaries (FSW Agility)
  - Periodic merge of FSW and Dictionaries into baselines
  - Early detection of Flight/Ground issues
  - Early training of mission ops team members
- Disadvantages
  - Constant request for feature changes
  - Sometimes too much flexibility can be a bad thing

Future Directions

- MTAK (MPCS Test Automation Kit)
  - An automated test scripting capability based on Python that can access any of the MPCS data and events, and trigger actions
  - Used to automate testing of “units under test” – not MPCS
  - Extensively used by MSL FSW developers and Testbed operators
- Advanced Visualization
  - Expansion of the current MPCS display capability, based on DISA standards and emerging standards

Contacts

- Michael Tankenson, MDAS Development Manager
  - (818) 393-1024
  - Michael.Tankenson@jpl.nasa.gov
- Lloyd Deforrest, MPCS Task Manager
  - (818) 393-5140
  - Lloyd.Deforrest@jpl.nasa.gov