The Space Congress® Proceedings

2000 (37th) Space Means Business in the 21st Century

May 3rd, 2:00 PM

Paper Session II-B - Checkout & Launch Control System

Jim Leondike
CLCS User Application Support Section Manager

Ken Clark
CLCS System Engineering Lead

Checkout and Launch Control System - Kennedy Space Center's Launching System for the 21st Century

Jim Leondike- CLCS User Application Support Section Manager
Ken Clark – CLCS System Engineering Lead

Abstract

The Checkout and Launch Control System (CLCS) is a NASA-led effort to design, develop, and implement a new Launch Processing System (LPS) at Kennedy Space Center (KSC). This re-engineering of hardware and software will support processing the Space Shuttle and future launch vehicles in the Launch Control Center (LCC), Hypergolic Maintenance Facility (HMF), Space Station Processing Facility (SSPF), Shuttle Avionics Integration Lab (SAIL) at Johnson Space Center (JSC), Dryden, and the Main Engine Processing Facility (SSMEPF).

CLCS is a distributed system utilizing state of the art technology in both hardware and software development. CLCS consists of four major systems: Simulation System (SIM), Real Time Processing System (RTPS), Business and Support Information Service (BASIS), and Shuttle Data Center (SDC). The goal of CLCS is to provide a system that will process the Shuttle more efficiently as well as reduce cost over the current system. To achieve these goals, the CLCS architecture will strive to provide standardization, rapid turnaround, automation, local control, multi-operation support, integrated displays, and desktop simulation using a flexible configured system.

Introduction

The Checkout and Launch Control System (CLCS) is being developed as the replacement of the Launch Processing System (LPS) at Kennedy Space Center (KSC). LPS was derived from the Shuttle processing requirements in June 1972, with the critical design review in September 1975. LPS is an integrated network of computers, data links, displays, controls, hardware interface devices, and computer software to control and monitor flight systems, Ground Support Equipment (GSE), and facilities used in direct support of Shuttle vehicle activities (CLCS, 1997). LPS first supported Shuttle Transportation System (STS)-1 in the Orbiter Processing Facility (OPF) in 1979, and continued support through the first flow leading up to the STS first manned orbital flight on April 12, 1981. CLCS is required to replace the existing functionality of LPS with modern off the shelf equipment. An LPS Upgrade Review Team was formed in April 1996, which recommended a National Launch Processing System (NLPS) in September 1996 emphasizing: Leverage technology and products, re-engineer the applications software, and employ rapid development. In November 1996 the NLPS project officially became known as CLCS, with funding approval the following month. This paper will discuss the CLCS subsystems, high level software components, and data flows used for commanding and monitoring shuttle and ground support hardware.

Goals

The goal of CLCS is to replace the outdated technology of LPS. Replacement parts are becoming more obsolete each year. Commercial peripherals must be modified to interface with LPS. The application software language used in the LPS (The Ground Operations Aerospace Language (GOAL)) requires unique training. Storage and processing capacities are saturated, as well as commercial development tools are not available for LPS. This use of old technology results in labor intensive operations (Hart, 1996). CLCS set out to utilize Commercial Off-The-Shelf (COTS) software and hardware to build a distributed, scaleable architecture to support launch, as well as new or emerging requirements for shuttle upgrades, experimenters, Space Station, and X programs. CLCS objectives include improving application fault tolerance for fail-safe operations, and consolidating data provisions to make more information available to control room personnel. The use of modern COTS hardware and software in CLCS will provide a flexible and more reliable system with increased support from vendors. CLCS shall also provide improved availability to the access of Shuttle data, reduce Operations and Maintenance (O&M) costs through commercial vendor support, allow the system to be upgraded with new technology, and allow the inclusion of new interfaces to support shuttle upgrades and new programs.

CLCS Architecture

Overview

CLCS consists of four major systems (figure 1). The Simulation System (SIM), Real Time Processing System (RTPS), Shuttle Data Center (SDC), and the Business and Support Information Service (BASIS). The Simulation System includes the math models to simulate orbiter, Solid Rocket Booster (SRB), GSE, External Tank (ET), and payload functionality. The RTPS is responsible for monitoring and commanding end items. The SDC is a repository for RTPS software and for data recording and retrieval. The BASIS is for accessing the Internet/Intranet, other KSC subsystems, office suite software, email, as well as monitor and retrieve data (CLCS, 1999). The BASIS provides Support Workstations (SWSs) and BASIS enabled office workstation connectivity and access to RTPS and non-RTPS applications. The BASIS is capable of monitoring data from the Shuttle Data Stream (SDS), retrieve shuttle data, reference documentation and drawings, simulation model control, and orbiter cable and connector information.

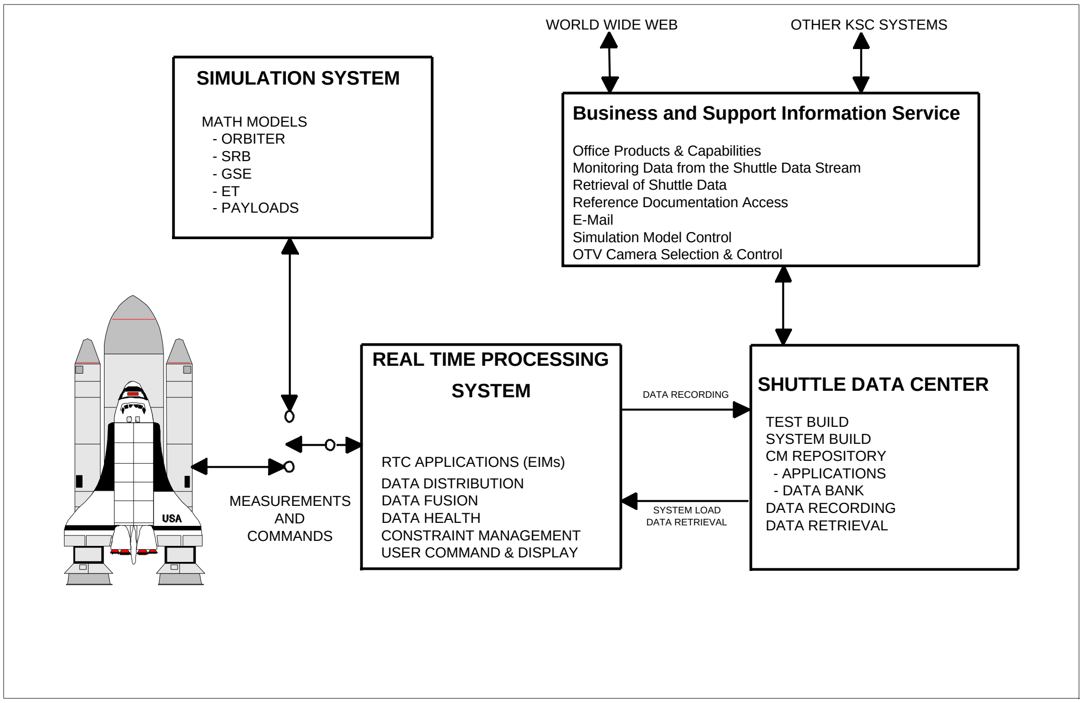


Figure 1. CLCS System Architecture.

Real Time Processing System

RTPS is a distributed system connected to four networks (CLCS, 1999) (figure 2). The Real Time Control Network (RTCN) connects the real time software and applications on Command and Control Processors (CCPs) and Data Distribution Processors (DDPs) to the gateways that interface with the Hardware Interface Modules (HIMs). The Display Control Network connects the DDPs and CCPs to the Command and Control Workstations (CCWSs). The Utility Network (UTN) is used for high volume data transfers (e.g. software loads), and the CLCS Inter-Set Network (ISN) links to the SDC for system loads and data retrievals.

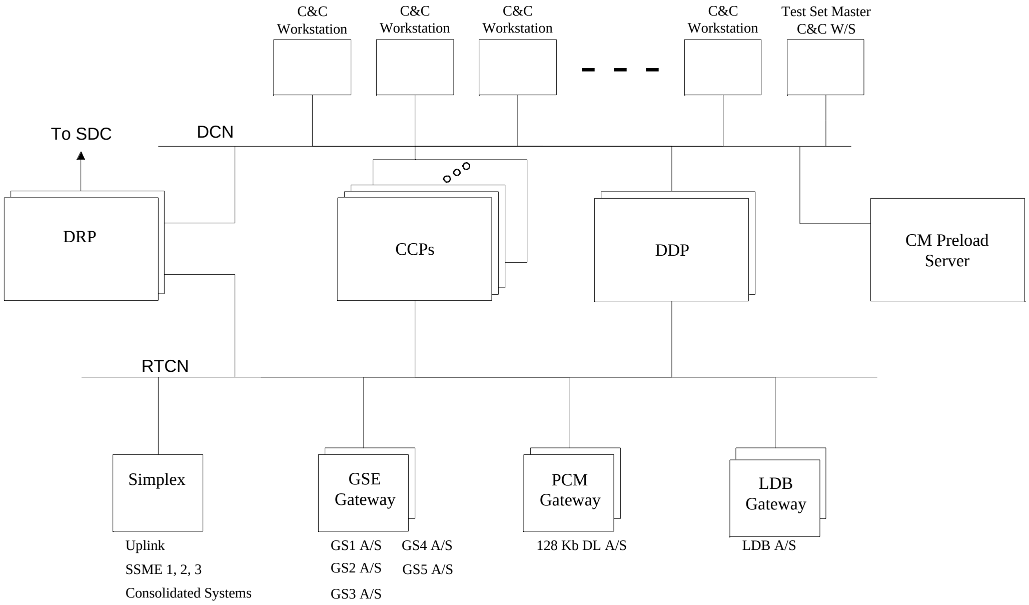


Figure 2. RTPS Physical Architecture.

The RTPS consists of four major subsystems: Gateways, DDPs, CCPs, and CCWSs. The Gateways are the replacements for the Front End Processors (FEPs). This subsystem is responsible for data acquisition, data conversion and calibration, change checking, Function Designator (FD) link status, and time stamping (figure 3). The DDP subsystem receives change data from the gateways at the System Synchronous Rate (SSR), which is 10 milliseconds (ms) (100 Hertz (Htz)). The DDP publishes the data to all the subsystems listening to the RTCN on the next SSR, and all the subsystems listening to the DCN each Display Synchronous Rate (DSR) which is 100 ms (10 Htz). The CCP subsystem is responsible for command processing and executing the Real Time Control (RTC) applications. The CCWS is the Human Computer Interface (HCI) to the system with user displays, command line interface, system viewers, and the safing system interface.

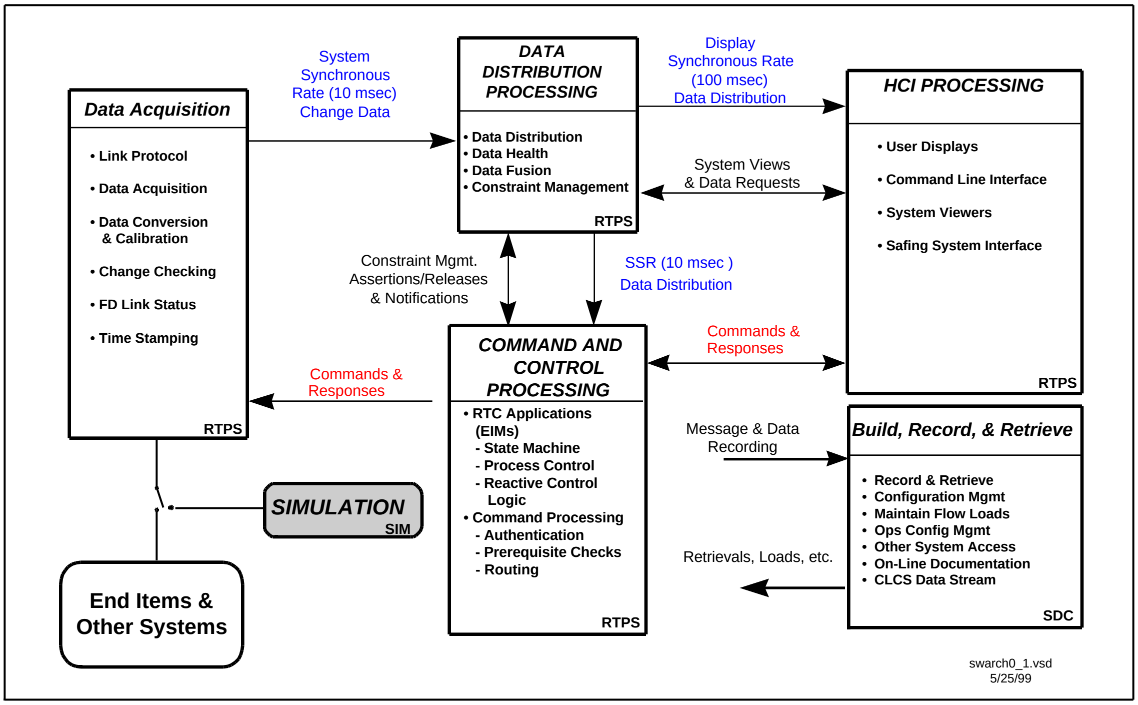


Figure 3. RTPS Functional Architecture.

RTPS Software and Data Flow Architecture

The Data Distribution Data flow is to acquire value and health data from the Gateways at the SSR, and merge the data into time order (figure 4). Data health algorithms are the combination of gateway status from each item of data and user determined health for the data item. Data fusion is the process of combining measurement data and other CLCS system parameters into a higher level of information including algebraic and logical manipulation of data. Health and fusion algorithms are performed on in-coming data and their results are published along with the gateway data to all the subsystems as FDs at the SSR rate on the RTCN and the DSR rate on the DCN. An FD is the name of a specific data item that is used as input to or is set by RTPS system or user application software. Measurement FD values may be acquired from end item hardware, RTPS system hardware, or created by RTPS system and user applications. Command FDs are stimuli to end item hardware or commands to RTPS system or application software. Each FD contains static information in the On-line Data Bank (OLDB) which is part of the Test Configuration Identifier (TCID). Static information contained in the OLDB includes items such as the FD's nomenclature, range, responsible system (RSYS), and detailed software information about that FD. The dynamic portion of the FD is stored in the Current Value Table (CVT) located on each subsystem. The CVT contains data such as FD Identifier (FDID), value, time of last value change, the FD's health, source, reason code, time of last health change, number of times data has changed, display attribute information, and data type information. The master CVT is located on the DDP, and copies are multi-cast to each subsystem.

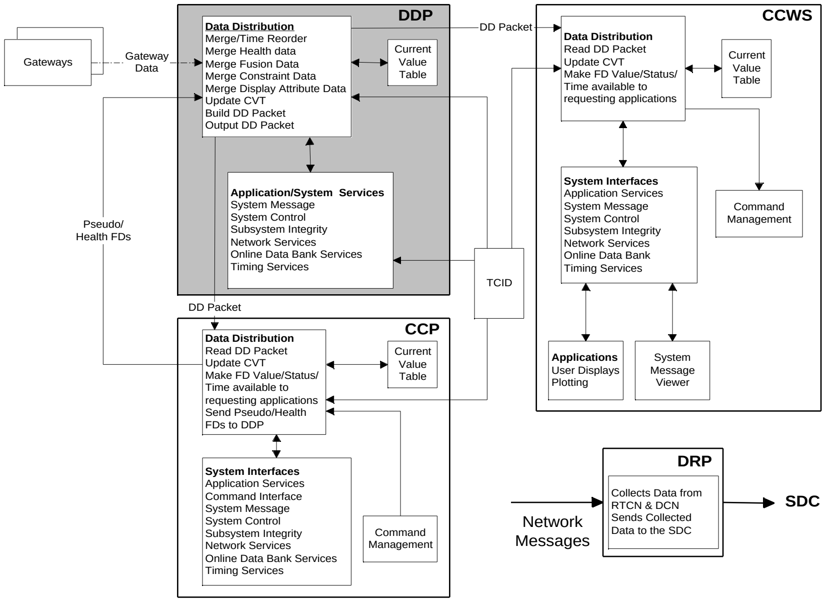


Figure 4. Data Distribution Data Flow.

The DDP subsystem is also responsible for constraint management (figure 5). This is the capability to monitor any FD for a predetermined condition and receive a notification when that condition is violated. These

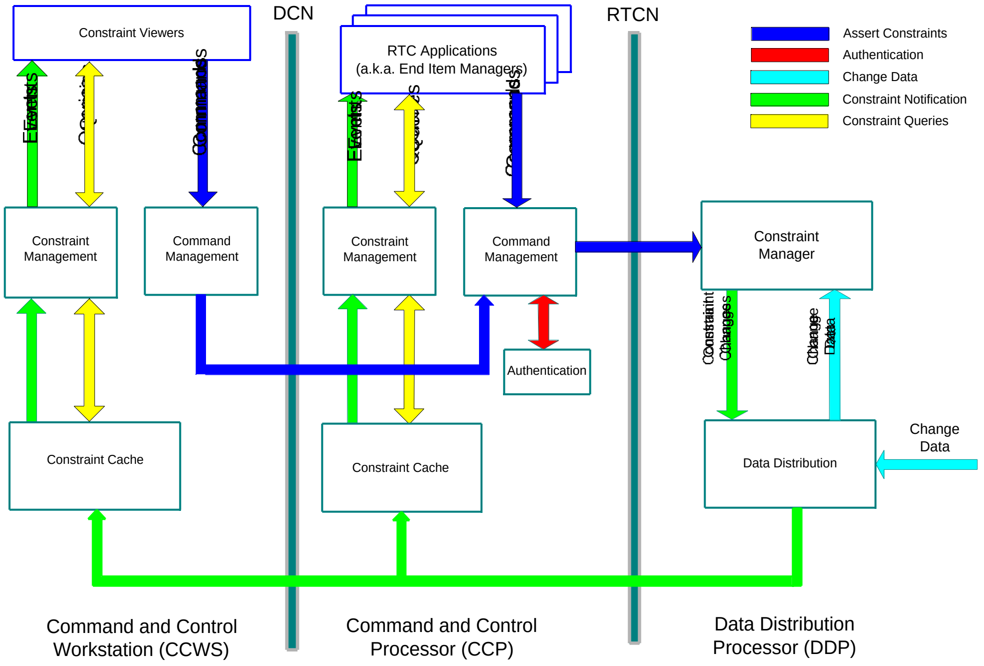


Figure 5. Constraint Management Data Flow.

constraint notifications are sent to real time applications and users sitting at a console. A constraint may trigger an application to react, may change a display attribute (display properties associated with graphical images), or start a Reactive Control Logic (RCL) sequence for hardware safing.

The CCP is responsible for monitoring the integrity of the system, managing commands, and hosting the RTC applications (figure 6). Command requests are sent from each of the subsystems using Reliable Messaging (RM) to Command Management running on a CCP. Command Management is responsible for authenticating the command to make sure the request came from inside the test set, and that the user has the proper commanding privileges. Command Management is also responsible to check if an end item has a Prerequisite Control Logic (PCL) sequence associated with the FD and execute the sequence to make sure the request will not damage flight hardware. Applications and users can command FDs, Constraints, Bus Terminal Units (BTUs), RTPS hardware and software, and the Orbiter's General Purpose Computers (GPCs). RTC applications run End Item Managers (EIMs) on multiple CCPs. RTC application software is the implementation of shuttle engineering requirements into software for checkout and launch operations (CLCS, 1999). EIMs are the primary command and control application of the RTC application software set to provide a centralized method for issuing end item commands and maintaining an accurate representation of the hardware state. EIMs contain End Item Components (EICs), which is a software module representing a hardware end item, and sequences.

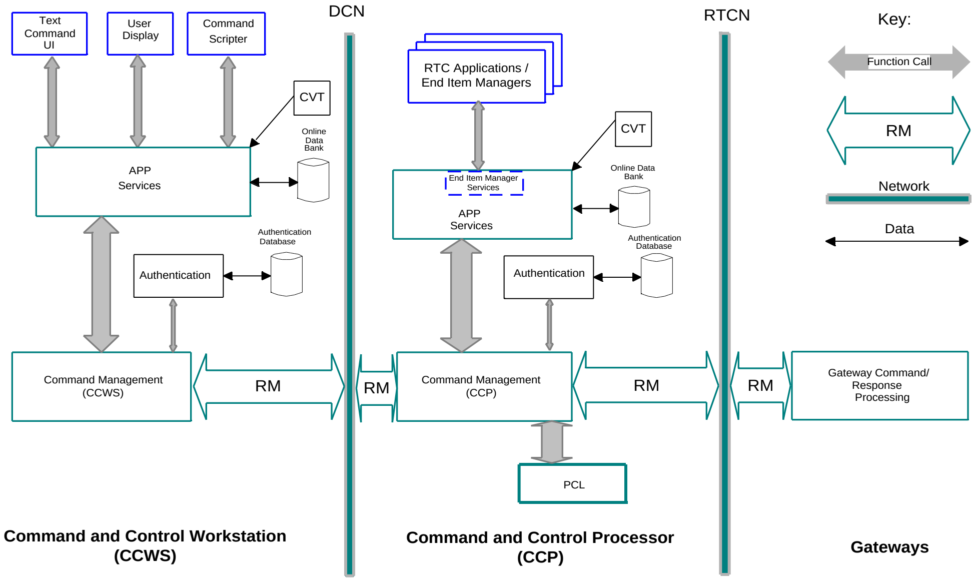
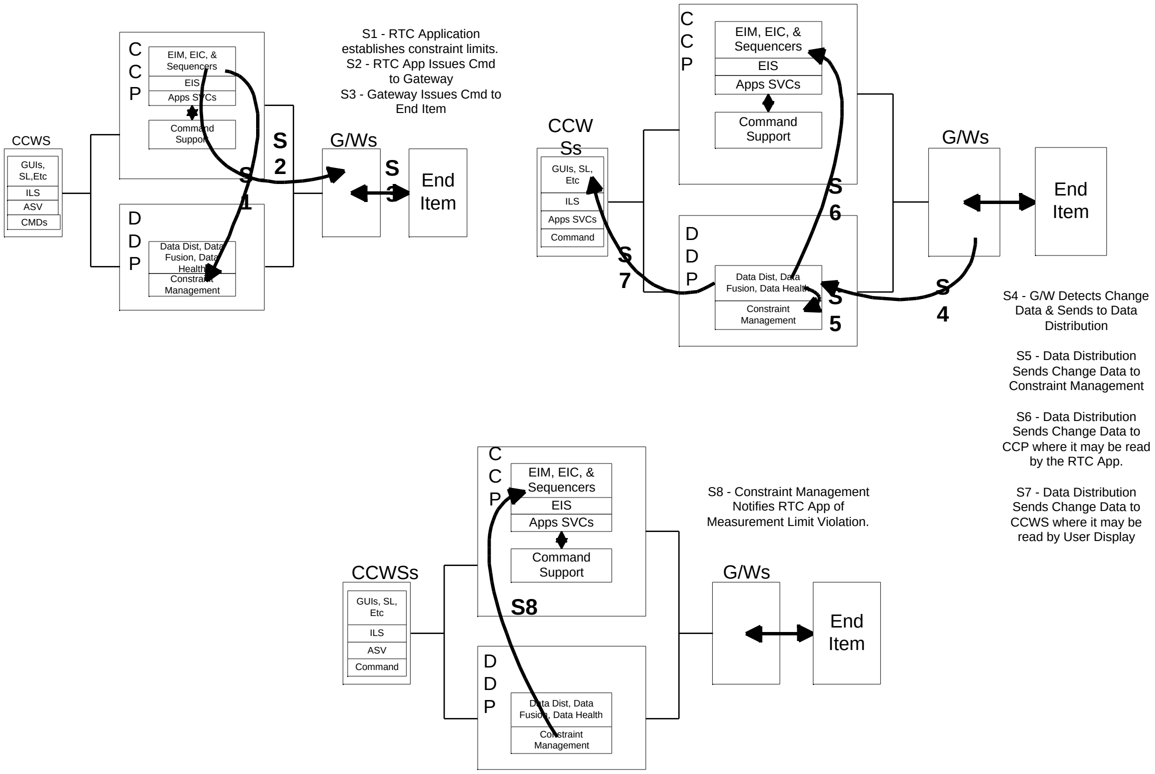


Figure 6. Command Management Data Flow.

The CCWS subsystems are graphical workstations, which allows the user at a console to monitor and command the RTPS. The CCWS's user interfaces consist of System Viewers and Test Application Displays (TADs) (CLCS, 1999). The Console Allocation Viewer allows authorized operators to allocate a user system to a CCWS. The Constraint Viewer provides system insight into constraints relevant to the system assigned to the CCWS. Display Monitor Viewer allows CCWS operators to specify a number of FDs that are of interest and view their current value. FD Details Viewer provides all details relating to a given FD. The Plot Viewer allows CCWS operator to plot one or more FDs. System Message Viewer gives the console operator a view to all the error, warning, and informational messages with in the RTPS. The System Checkpoint Viewer allows O&M CCWS operator to view messages of interest. The System Status Viewer provides details relating to the current status of all subsystems in the test set and to relevant information into hardware and software inside the subsystem. Command Processor allows manual commanding from the CCWS. TADs are the primary user interface for monitoring all vehicle and GSE data. TADs can also issue manual commands to hardware end-items, and control automated sequences.

The Real Time Control Loop consists of applications running on the RTCN to monitor data and constraints, and execute commands (figure 7). This consists of data from the gateways, DDP, and CCPs.



S1 - RTC Application establishes constraint limits.
S2 - RTC App Issues Cmd to Gateway
S3 - Gateway Issues Cmd to End Item

S4 - G/W Detects Change Data & Sends to Data Distribution

S5 - Data Distribution Sends Change Data to Constraint Management

S6 - Data Distribution Sends Change Data to CCP where it may be read by the RTC App.

S7 - Data Distribution Sends Change Data to CCWS where it may be read by User Display

S8 - Constraint Management Notifies RTC App of Measurement Limit Violation.

Figure 7. Real Time Control Loop

The Manual Control Loop (figure 8) is the user interaction with the RTPS when sitting at a console.

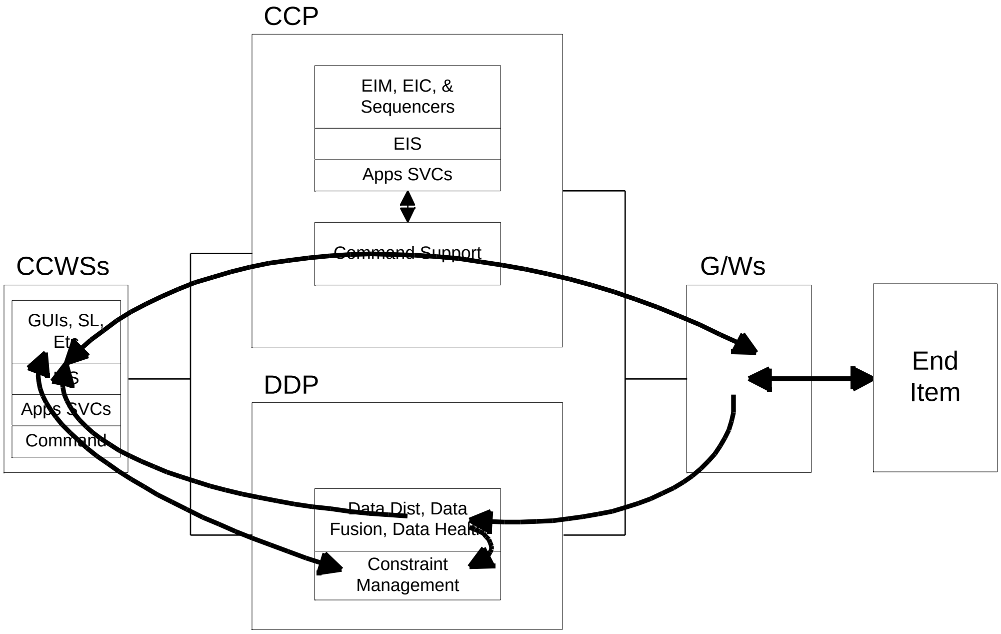


Figure 8. Manual Control Loop.

Summary

CLCS is being built to replace LPS with new technology. For rapid turnaround and early user involvement, CLCS was to evolve through 10 deliveries. Designing and integrating the real time and manual control loops in early deliveries, delivering operational GSE hardware and software at the HMF, then supporting flow in the OPF, with completion when CLCS supports launch. The CLCS is composed of the RTPS, SDC, SIM, and BASIS. The RTPS provides the capability to monitor and command elements of flight hardware and GSE (CLCS, 1997). The SDC is the repository for the test data and provides the capability to build test packages for the configuration of the RTPS. The SIM gives RTPS software the capability to debug and certify their applications against math models. The BASIS provides CLCS SWS connectivity and access to non-RTPS applications and data. COTS applications are used whenever possible, and if custom applications need to be built, then a high level software language is used for a higher degree of portability between platforms. COTS hardware is also utilized whenever possible. This strategy provides a reliable system that is modular, expandable, and extensible. CLCS is based on open hardware and software standards, easily incorporates new technology and user developed applications, and provides inherent user interface improvements.

References

CLCS, June 1999. Checkout and Launch Control System Critical Design Review.
CLCS, October 1997. CLCS Employee Orientation Guide.
Hart, Retha , August 1996. Why NASA Needs to Upgrade the Launch Processing System.