

# GROUND/MAN-MACHINE INTERFACES FOR ORBITER CHECKOUT

F. Herb Blackmon  
IBM Test & Operations  
Kennedy Space Center

## ABSTRACT

This paper discusses the challenges presented by the concept of a reusable, cargo carrying space vehicle, and how those challenges were met for the Space Shuttle. Areas discussed here include the complexity of the vehicle, the ground support system, the onboard computer system, ramifications of a reusable vehicle, and the turn-around objectives for Shuttle flights.

After six successful flights at the time of this writing, it can be safely said that the challenges presented here have been basically met.

## INTRODUCTION

Adjacent to the Vertical Assembly Building (VAB) at Kennedy Space Center (KSC) is an APOLLO vehicle, complete with all the stages required to orbit the Moon and return. The long, sleek and sharply pointed configuration could be imagined to be a spear used by some mythological god to fling into the heavens to bring down a passing star.

In the VAB, in preparation for a Space Transportation System (STS) flight is a configuration of two Solid Rocket Boosters (SRBs), an External Tank (ET), and an Orbiter. Unlike the APOLLO launch configuration, the STS looks bumpy, unsymmetrical, and no god would have any use for it. Indeed, the STS configuration could be thought of as the bumblebee of space flight - it appears to be an ill-conceived design, but does its job beautifully.

Although the appearance of the two vehicles is remarkably different, the technical aspects of preparing each for, and accomplishing a launch are even more different. This article will present some comparisons in vehicle complexity, ground support systems, objectives, and turn-around requirements to demonstrate how the STS challenges were responded to.

## VEHICLE COMPLEXITY

As stated earlier, the appearance of STS is very different from its predecessors in America's space ventures. The STS is several orders of magnitude more complex than earlier launch vehicles. Indeed, STS is the first vehicle conceived, designed, and implemented strictly as a space transportation system. Earlier vehicles used to put payloads into space were, at best, quasi-military in concept and design, and in some cases such as Thor, Atlas, Delta, etc. were military vehicles adapted for use by NASA.

Several items may be compared between APOLLO and STS to demonstrate the increased complexity of the vehicle: engines, control capability, computers, payloads, and landings.

## Engines

The APOLLO vehicle, although consisting of three stages, can be thought of as a single engine vehicle, since only one type of engine was ignited at a time. Even in the Saturn V first stage with its five engines, the engines were arranged symmetrically about the center of the vehicle with the middle engine fixed.

The STS, in a nominal launch, also has five engines ignited. However, two of these are SRBs, while three are the Space Shuttle Main Engines (SSME); and the engines are not symmetrical about the centerline of the total vehicle.

The SSME's are liquid fueled from the ET and controlled by their own computer (Engine Controller) which interfaces with the onboard computer system, while the SRBs are solid fueled and controlled directly by the onboard computer system.

In some non-nominal situations, the STS can also utilize the two Orbital Maneuvering System (OMS) engines to assist in later stages of the launch. These engines are liquid fueled from tanks onboard the Orbiter.

### Control Capability

In APOLLO, the astronauts were "along for the ride" - that is, the launch and ascent were automatically controlled by the ground and onboard computer. The crew had very few options if trouble developed during the ascent into Earth orbit.

The STS has both automatic control and manual control. The automatic control is similar to APOLLO. The manual control allows the pilot to manually steer the vehicle using the Rotational Hand Controller (RHC) from SRB ignition through SSME shutdown.

The STS also has the capability, for non-nominal situations, of several abort modes: a Return-to-Launch-Site (RTL), a Trans-Atlantic Landing (TAL), an Abort-Once-Around (AOA), and an Abort-to-Orbit (ATO).

### Computers

The APOLLO configuration contained a single computer in the Instrumentation Unit which controlled the vehicle's flight. The computer was preloaded with the flight profile, data, etc. and, for ascent, had no crew interface capability.

STS, on the other hand, has five General Purpose Computers (GPCs) which control the vehicle's flight and provide interfaces to the crew to allow them to make inputs during ascent. Four of the computers run in a "redundant set" using the Primary Avionics Software System (PASS), while the fifth is loaded with the Back-up Flight System (BFS). During ascent the four PASS computers synchronize themselves over 300 times per second and pass data to the BFS computer to allow it to track the flight. Should the PASS set fail, the crew can order the BFS computer to take command and provide for a safe flight to either continue or return.

### Payloads

APOLLO was designed to put a single payload into space - the crew compartment (and Lunar Excursion Module) and its contents. It was strictly an exploration program to increase our knowledge of near space through manned exploration.

STS is a service system designed to put many types of payloads into space. With its Remote Manipulator System (RMS), or "arm", STS can deploy and retrieve payloads, shuttle them back and forth, and perhaps even effect repairs on faulty payloads in orbit.

### Landings

The APOLLO crew compartment effected a ballistic reentry with a water landing where it was plucked from the water by a ship stationed in the landing zone, then transported back to the USA for display.

The STS lands in a conventional aircraft manner on a landing strip using conventional areosurface controls such as elevons, rudder, and speedbrake, with retractable landing gear. It can then be readied for the next launch.

## GROUND SUPPORT SYSTEMS

The ground support systems of APOLLO and STS are as different in nature and complexity as the vehicles themselves are. Indeed, the consoles of the STS ground system are said to be inverted APOLLO ground system consoles!

### Apollo Ground Support System

The APOLLO ground support system was composed of two ground computers which were transistorized, large scale, serial, digital computers. One computer was located in the Launch Control Center (LCC) and communicated via a data link with the other computer in the Mobile Launcher.

The two computers were required to be on-line and operational for checkout and launch so there was very little redundant or backup hardware. There were numerous single-point failures, and nearly 50 percent of the hardware was located within a few hundred feet of the vehicle, creating hazardous areas for maintenance.

More than two hundred people were required to monitor the meters, lights, plotters, etc. used in the launch complex to checkout and launch the vehicle. Data was obtained through fixed telemetry streams and fed into the LCC for interpretation by the individual engineers.

Testing of the launch configuration was largely a serial operation. A particular subsystem would have to be fully tested before the next subsystem could be started.

The net effect of this system was a long, tedious process to ready the launch vehicle for flight.

### STS Ground Support System

The STS ground support system, termed the Launch Processing System (LPS), is designed to meet the rapid turn-around requirements for a reusable vehicle and to reduce the number of people involved in vehicle testing and launch. At the same time, the LPS preserves the test engineer's direct control over checkout and launch procedures.

The LPS is a network of minicomputers (up to 64) which share a common data buffer to pass data and commands between computers and the vehicle. Communications between the LPS in the LCC and the vehicle is accomplished via a Launch Data Bus (LDB) which is "attached" to the GPCs onboard.

Two key differences between APOLLO and STS have been realized through LPS: visibility through CRT displays (and thus digital display in engineering units of data) of status and data, and the ability to perform parallel testing.

Also realized through LPS, although not readily "seeable" to the user, is a high degree of flexibility and redundancy. Since the LPS is a distributed system, any minicomputer can be loaded from a master computer with a selected subsystem load. Thus, if a console and/or a computer fails, a spare console/computer can be quickly brought up in its place. Critical system functions are backed up in computers in such a manner that a switch can be made with minimum loss of data.

An overview of the LPS and its architecture is presented in "The Launch Processing System for STS and DoD Space Shuttle" by Don G. Satterfield (IBM), published in the IBM/FSD magazine Technical Directions, Fall/Winter 1981, Vol. 7, Number 3.

### OBJECTIVES

The objectives to be met in producing the ground/man-machine interfaces for Orbiter checkout were:

- \* Rapid turn-around - Initial requirements for an operational STS called for a 160 hour turn-around (10 working days of two shifts each).
- \* Minimization of "fixed" software - The amount of software residing in the GPCs uniquely for checkout was to be minimized as well as minimizing the ground software which was "fixed", that is, tailored for a specific vehicle/payload configuration.
- \* Flexible capabilities - Provide engineers with "menu" selection of predefined tasks to be done and provide the capability to perform contingency tasks.
- \* Complete vehicle checkout - Provide the capability to test the Orbiter in the Orbiter Processing Facility (OPF), the STS (Orbiter, ET, and SRBs) in the VAB and at the pad, and the payload(s) at the Cargo Interface Test Equipment (CITE), OPF, and pad.

### IMPLEMENTATION

The objectives spelled out have been met through implementation of computer programs in onboard GPCs in PASS and in the LPS computers, and procedures for engineers to use.

### Onboard Capabilities

The primary interface between the LPS and the vehicle is the Launch Data Bus (LDB). While a direct mode between LPS and the command decoders onboard exists, we are primarily concerned here with the mode where the LPS and the GPCs communicate. In this mode, polling on the LDB occurs between the LPS and the GPCs every 40 milliseconds, with an established protocol being followed.

Commands from the LPS may be directed to one of six "functional destinations" in the GPC and may range from simply setting a discrete on in an MDM to initiating a series of commands to an actuator in a predefined pattern up to 100 times a second for a Frequency Response Test (FRT).

Single function commands are sent through "operators" in either the Test Control Supervisor (TCS) mode, or the System Avionics Command Support (SACS) mode, based on availability determined from the Operational Sequence the GPC is in.

Functions such as the FRT reside onboard in Explicitly Coded Programs (ECPs) which are designed to accomplish specific tasks. These are requested from the LPS via a single operator. The ECPs, once requested from LPS, will execute, based on parameters from LPS contained in the request, without further LPS intervention.

The PASS also uses the LDB to communicate with the LPS in the final phases of a launch countdown. The LPS Ground Launch Sequencer (GLS) uses the LDB to send control commands to the onboard Redundant Set Launch Sequencer (RSLS). LPS monitors Launch Commit Criteria (LCC) on the ground from the STS and RSLS to determine the status of the countdown.

Another key element of vehicle checkout is the securing of data from the various subsystems of STS. This is accomplished in PASS during vehicle checkout by the Housekeeping Data Acquisition (HDA) function which reads the various avionics related hardware and provides the data via downlist to LPS. The type of data and its format is selectable in predefined formats. So, unlike APOLLO with its fixed telemetry stream, STS has selectable telemetry streams with selectable downlist data imbedded in it.

Finally, the PASS has several autonomous vehicle checkout capabilities which are usable from the onboard keyboard/CRT system. These are called SPEC (Specialist) functions, and provide for predefined test and checkout capabilities.

#### Ground Capabilities

The LPS, as discussed earlier, is composed of a network of minicomputers. Each console/computer can support three operators who may be doing testing of different aspects within their subsystem in parallel.

In addition to the computers at each console, minicomputers are used to interface to the LDB and to receive data from the STS. These computers are called Front-End Processors (FEPs) and perform such functions as preconditioning data, formatting requested commands properly for the LDB, switching data formats, etc.

Engineers produce programs for the console computers using the Ground Operations Aerospace Language (GOAL), written by IBM for use in LPS. This language is oriented for engineering terminology and provides powerful, yet flexible capabilities.

In addition to the predefined capabilities generated through GOAL programs by the engineers, LPS contains a Command Processing system whereby an engineer can "build" commands to be sent to STS via the LDB. This capability allows the engineer to react to anomalous conditions and/or to accomplish a minor task which would not warrant generation and validation of a GOAL program.

#### ENHANCEMENTS

In large, complex computer systems, initial designs are seldom completely adequate or satisfactory. The STS checkout system is no exception to this. Many enhancements have been suggested and are either already implemented or planned.

Drivers for these enhancements, for the most part, have been the reduction of "special" considerations, greater flexibility, and increased visibility. For example, some enhancements that have already occurred are:

- \* The selection of aerosurface limits for movement when the Orbiter is in a horizontal position in the OPF or a vertical position at the pad. Initially these limits had to be "patched" in the PASS to change them. Now, an entry on a SPEC function display allows selection.

- \* Increased communication of TCS status between PASS and LPS. Additional status parameters have been added to ECPs to provide the LPS better visibility into reasons for rejection or failure of ECPs.

\* Selection and saving of IMU parameters. Similar to the aerosurface limits, several IMU calibration parameters change when the Orbiter is in different locations. Originally, the only way to change these was via a patch to PASS. Now, on the IMU Calibration SPEC, the vehicle site can be selected prior to calibration. Also, the IMU checkpoint data can now be saved on all areas of each Mass Memory Unit (MMU) directly from the SPEC function.

The enhancement of ground/man-machine interface is not complete. Indeed, a task force of engineers from the Space Shuttle community still meets periodically to consider suggested enhancements which will make the system faster, more reliable, and easier to use.

#### SUMMARY

The STS is a very complex system which required a quantum step forward in the ground/man-machine interface system for checkout. This step has been made with the combination of the LPS and the PASS, so that the resulting system is:

- \* Flexible
- \* Powerful
- \* Conducive to parallel testing
- \* Highly visible
- \* Conducive to multi-vehicle flow testing.

#### ACKNOWLEDGEMENTS

The author wishes to express his thanks to Tom Santos, Elliot Burton, and Don Kleckner (all IBMers who were at KSC during the APOLLO era) for their support in producing this article. Also, to the IBM Test and Operations group at KSC for reviewing the article to make sure I didn't stray too far from the path.