By Michael Meagher,
Software Engineer,
SL Corporation.

Kennedy Space Center Upgrades Launch Control Systems for the 21st Century

One of the grandest endeavors ever undertaken by humankind is that of manned space flight. The world's most renowned spaceport is Americas Kennedy Space Center (Cape Canaveral, FL); which launched legendary space travelers of the Mercury, Gemini, and Apollo programs. The National Aeronautics and Space Administration (NASA) currently maintains a fleet of space Shuttles for all its manned missions. These Shuttles will be heavily utilized for the construction and servicing of the International Space Station (ISS). The space station will serve as an orbiting laboratory and serve as an outpost for future space exploration. Since much of the technology installed in Kennedy's Launch Control Center (LCC) dates back to the early days of the U.S. space program, an effort is underway to modernize critical launch control systems with an eye toward safety, ease of use, upgradability and cost effectiveness. This article will examine some of the innovative new technologies being deployed and their impact on launch control activities.

TECHNOLOGY OVERHAUL FOR ISS SUPPORT

The monumental task of preparing a Shuttle for a mission calls for a variety of important operations to be performed by NASA and contract personnel. Engineers conduct many of these operations from the Launch Control Center, a large concrete structure three miles from the launch pads, constructed during the Apollo program. The LCC contains three separate firing rooms; each organized into clusters of computers, which control specific operations such as tanking, hazardous gas monitoring, payloads processing, avionics and communications. From these computer consoles, NASA engineers issue the commands that prepare man and machine for the long journey into space. Teams of highly skilled engineers rely on computers to assess and control the Shuttles many complex systems.

While still functional, much of the hardware and software currently used in the LCC has not been upgraded since the late 1970s, when NASA replaced the Apollo launch system with the current Shuttle launch system. Despite great leaps in computer technology over the past 25 years, many of the old LCC computer systems have remained in place. NASA management has recently mandated that the nation's fleet of four space Shuttles will remain operational until at least the year 2012, in order to serve as the primary launch vehicles for deploying and servicing the international space station. Launching of the space station will require many Shuttle missions over the next several years in order to transport the equipment and personnel required during the construction phases. Once constructed, the station will require frequent deliveries of fuel, provisions and personnel rotations. The multi billion-dollar space station effort is dependent on a reliable launch vehicle while much of the technology used to service the Shuttle is aged.

The primary objective of the LCC modernization program is to improve the technologies involved in check out and launching operations so that Shuttle launchings can safely continue well into the 21st century. This will protect NASA's investment in the ISS by ensuring viable launching operations. In addition, the systems put in place must be flexible enough to handle new components and potentially new launch vehicles, such as the X-33.

UPGRADED GRAPHIC DISPLAYS IMPROVE EFFICIENCY

A modernization program will help boost the productivity of NASA engineers by allowing the launch control systems to benefit from recent advances in computer hardware and software technology. Traditionally, engineers in the Launch Control Center have relied on custom software written in Ground Operations Aerospace Language (GOAL) which runs on a custom Modcomp based consoles. While still effective, the older systems require highly specialized maintenance and support. Obtaining replacement parts for such aged computer equipment is often difficult and expensive. The current system will remain active until completion and delivery of the new Checkout and Launch Control System (CLCS) in 2002.

In the older system, graphic images bear little resemblance to the actual systems being monitored. The character-based graphic displays, which are built in code, are expensive to maintain. Engineers conducting launch activities must wade through many screens of textual information in order to monitor critical launch data. Sensor readings are listed by their tag number and organized into rows and columns of textual values. These older system graphics provide little if any information about the location of a particular measurement. Although it performs adequately at real-time rates using custom assembly code, the computer hardware is obsolete when compared to modern high performance workstations. The graphics capabilities of these character-based systems are very limited in light of what is possible using graphics toolkits on current graphics windowing systems.

In 1996 NASA set its sights on improving to the quality of the information being displayed to the launch teams while keeping an eye on the expense involved in implementation and support of the new system. The initial focus of the engineers was a product evaluation to identify the best tools to fulfill the requirement for more sophisticated real-time graphic displays. A major concern in the selection process of a real-time graphics tool was the cost incurred during display creation. In addition, the graphics need to be easy to maintain, since this large-scale systems integration project involves hundreds of engineers. The search led to the selection of SL-GMS from SL Corporation (Corte Madera, CA) as the graphics toolkit for building the new front-end real-time graphics displays for the CLCS application. SL-GMS was chosen primarily for its object-oriented architecture, which allows for component reuse and interoperability with other system software, such as data acquisition modules. The drawing tool from SL Corporation gives engineers the ability to quickly construct custom real-time graphics displays.

Once constructed the SL-GMS application program interface links an object's graphical appearance to the underlying application data, which is continually changing during checkout and launch activities. New screens can be developed and plugged into the runtime system without any modification or reprogramming of the runtime application. This approach reduces the number of lines of code that must be developed -- and fewer lines of code require less maintenance. This solution fit neatly into NASA's "better, cheaper, faster" mandate. In addition, SL-GMS runs on a variety of computer hardware (RISC, Alpha, INTEL) and operating systems (UNIX, VMS, NT), freeing NASA from the constraints and risks of a single-vendor hardware system.

OBJECT ORIENTED ARCHITECTURE SIMPLIFIES DISPLAY DEVELOPMENT

One objective of the new system is to provide an intuitive and informative view of pre-launch operations in order to conduct safe launch system processing. In order to fulfill this objective the new systems being implemented at the LCC capitalize on commercial-off -the-shelf (COTS) hardware and software. The graphics systems are housed in an operator console with a standard keyboard and mouse. The new computers are primarily UNIX workstations connected by a high capacity Ethernet to data distribution computers. Development work is being conducted on both Sun Ultra and Silicon Graphics O2 workstations, while deployment platforms will be primarily Sun Ultra Sparcs. Commanding is initiated from Command and Control workstations (Unix) via a common object request broker (CORBA) interface. In addition a display-only Windows NT version will be available to personnel for remote monitoring capability. All of the Command and Control workstations are connected via a high bandwidth Ethernet to Command and Control Processors, Data Distribution Processors and Gateways. Monitor-only data is also provided to office workstations located throughout KSC, other NASA centers and supporting facilities throughout the United States.

In order to better monitor and control a variety of subsystems related to pre-launch activities, new front-end displays are being constructed using SL-GMS. The new graphics displays provide launch engineers with a realistic view of system status. Custom graphics displays are built using a menu driven graphical editor (GMSDRAW), which allows dynamic properties to be associated with the graphical objects. The graphical attributes can then be associated with the correct driving data point. The completed model is then saved to disk and checked into a revision control system. A runtime application loads the necessary graphics and links them to real time data. Object oriented data acquisition tools are combined with the graphic objects so that the graphics change their appearance in response to real-time vehicle data. The new displays allow operators to quickly ascertain overall system status and sub-systems health. New displays have been built to monitor and control systems such as tanking, vehicle health and payload readiness. Not only does the new system alert operators of alarm conditions, the new displays also show operators the location of the problem.

Since the graphics represent the physical layout of the equipment being monitored, operators can quickly ascertain the location of an alarm condition and observe which systems might be the cause of the anomaly in order to take corrective action. An example would be a sudden drop in tank pressure. Such a condition would be easily recognizable on the screen as the fill color of the tank graphic changes to a color such as blinking red. This color change would be clearly visible in order to capture engineer's attention. Since the graphics portray the actual hardware, cause and effect relationships are readily apparent. Additional screens can be brought up and corrective actions taken quickly. Commanding is initiated by clicking on the graphical representation of the hardware to be controlled. A popup menu is invoked and the appropriate commands can be issued by simply clicking on

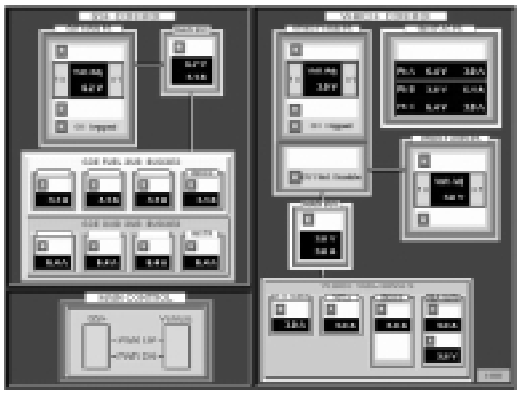

Figure 1.

a push button.

The new 64 bit UNIX Sun Ultra Sparc workstations running the X windowing system provide state-of-the-art graphics, which render the SL graphics in near real-time. The newer graphics better depict the physical systems they are monitoring in order to provide an intuitive view of the Shuttles many complex control systems. Some of the graphics screens are organized in a fashion similar to the older graphics system, with rows and columns of data changing value as the data updates. Even in these primarily textual screens, a more logical organization of the data is being implemented. For instance, all of the oxidizer data points are located in the same portion of the screen, unlike the older system, which simply listed the data in a sequential order. If alarm conditions are reached, the text will change color to indicate a warning or alarm status.

The new graphics displays more accurately communicate critical launch data to engineers in the Launch Control Center by representing the physical interrelationships of the hardware being monitored. Shuttle engineers have already built many of the initial graphics screens to meet specifications for initial milestone deliveries. The new system was tested during super lightweight tanking operations for Shuttle flight STS-91 June 1998. NASA used the new graphics system to support the operation of a new lightweight external fuel tank. In order to boost the cargo carrying capability of the space Shuttle, the new super light weight external fuel tank weighs 7,500 pounds less than the previous model. This reduction in weight will allow the Shuttle to carry heavier payloads and reach the higher orbit necessary for servicing the space station. Informative graphic images integrated with a real-time data collection system provide engineers in the Launch Control Center up to date and accurate depiction of operational systems.

INTEGRATION WITH DATA ACQUISITION AND CONTROL SYSTEM

The current systems upgrade includes functions such as data acquisition, data validation, and data transmission, data conversion and distribution. Physical measurements such as temperature and pressure are monitored at critical locations. Data from Ground Support Equipment (GSE) is received via Hardware Interface Modules (HIMs), Flight hardware (Orbiter, External Tank, Solid Rocket Boosters) data is received via hardwire (Launch Data Bus) or through telemetry. Each data measurement is given a unique identifier, referred to as a Function Designator. The data is then broadcast to the LCC and is written into shared memory in each individual Command and Control workstation. Engineers in the LCC can then view the information by invoking a viewer program on their workstations. The runtime graphics application reads shared memory and updates the screens accordingly so that the graphics reflect the most recently distributed data sets. By simply clicking on the appropriate push button, engineers can select and monitor specific displays. As new screens are activated, new sets of variables are registered with the data distribution engine. The data server then sends updated data sets to the workstation as remote sensor data changes. Custom graphic displays change their appearance in response to this real-time data.

Figure 2.

In addition to monitoring, commanding functionality has been developed which facilitates the control of hardware components such as pumps and valves. Manual control functionality is achieved using the SL-GMS package in conjunction with a CORBA interface to issue commands to physical devices. The user must first initiate commanding screens by clicking on the graphical representation of a specific component. Upon authentication of the user's authority, a popup window is activated which contains the push buttons for commanding the selected device. Commanding operations are accomplished via callback functions associated with graphic input objects. Commanding from SL screens is conducted from radio-style check boxes which, when selected, call a function which initiates a CORBA communication. This CORBA command is first sent to the appropriate Command and Control processor where it is again authenticated, converted to a computer-to-computer message, embedded in a packet and transported via a reliable messaging service to the Gateway. The Gateway then sends the command to the appropriate physical device, such as programmable logic controller (PLC) to carry out the specified operations (i.e. "open a valve").

BUILDING NEW REAL-TIME DISPLAYS

The graphical user interface (GUI) built using SL-GMS now provides active graphic representations of launch system components. Graphic objects (models) are built using the menu driven GMSDRAW tool where dynamic properties are applied to objects so that they change their appearance in response to real time data changes. Simulated data can be fed into the graphic image to test its dynamic behavior. Completed graphic objects are saved to disk as model files. Top level graphics screens can be assembled by dragging and dropping sub-model graphic components from a palette and assigning the correct driving data from a list of function designators. Collections of sub-models are organized into palettes so that other screen builders can reuse the components. The simplicity of the point and click approach to screen building saves

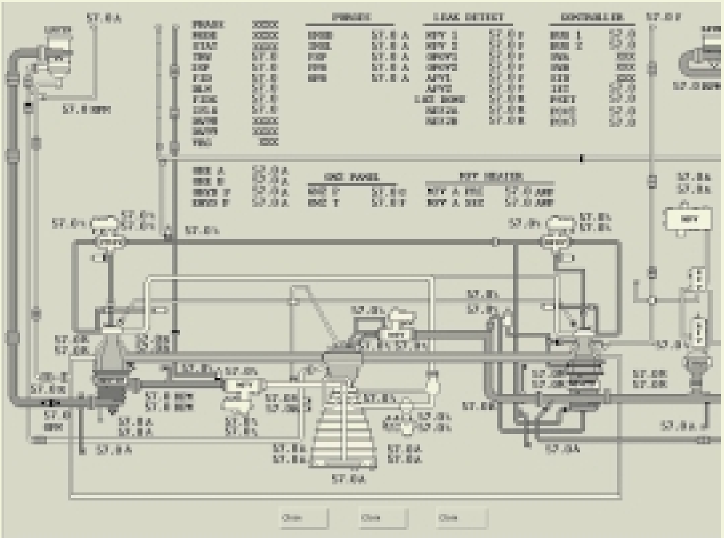

Figure 3.

valuable application development time. Screens put together in the prototyping stage can be used in the actual runtime system upon completion of validation testing. Modification of screens can be easily accomplished using the SL-DRAW tool rather than modifying application code..

Engineers have built graphical representations of the Shuttle's control panels using the GMSDRAW tool. Many of the pre-launch checkout operations are conducted in Orbiter Processing Facilities (OPF) where sub-systems such as landing gear, brakes, and the remote manipulator arm must be tested prior to each launch. Now, intuitive tests can be run and viewed in an interactive manner. New computer graphics screens show operational test results to engineers in the LCC as the tests are conducted on a Shuttle in the OPF. Examples of such tests include graphic depiction of the Space Shuttles' landing gears, which retract and extend as the result of changing MSIDs (Measurement/Stimulus Identifier). When testing the remote manipulator arm, parameters such as voltage or current flow are monitored on computer screens. Graphs are used to show historical data values. Expected results can be graphed against actual test values to signal any deviation in the test data.

Once built, models are loaded into an SL-GMS runtime application, which binds the graphic model to the appropriate live data sets. The object-oriented technology of SL-GMS extends to the programmatic structure of the HCI as well. Code modules are organized into higher level objects, called "screen states," which encompass a set of graphics (models), a set of event handlers (methods), a set of state parameters (state flags) and a set of application variables which are relevant to a particular graphic. Once the runtime application is launched, the click of a button or icon can activate new screen state. Users have the ability to print hard copies of screens by clicking on a button, allowing for off-line analysis and trouble shooting.

Using a state based framework application code is compact and efficient. Isolating problems is simplified, since the application is written in a modular fashion. A variable initialization function is used to register a list of variables, which drive a particular model. The runtime application then makes a request to the data distribution system so that new values will be posted to local shared memory. Using this approach, new models can be loaded into the runtime application, which automatically connects them to the proper data. New graphics can be constructed in the graphics editor and no re-coding is required to attach the graphics to an active data set. When new data is posted to the local workstation, graphic displays update their appearance in a manner specified by their dynamic properties. Graphics change their attributes in order to reflect the latest FD's values. Dynamic object properties, such as fill color, text-content or text-color can change to reflect alarm conditions. Graphic representations of tanks can be filled or emptied to reflect actual operating conditions. Valves change their appearance to portray their current settings.

Any Command and Control Workstation in the LCC can pull up any relevant screens, keeping engineers well informed of any related systems' status. When a screen state is no longer being viewed, it can be deactivated and the associated memory freed. The data acquisition system is then notified that the application no longer requires information about a particular set of variables. By "scooping," the active data set application efficiency and performance is boosted. In many cases, identical graphics may point to unique data sets. For example one model of an engine could be used to reflect the operational status of any of the Shuttles main engines. In this case a main engine model can be instanced and driven by a unique data set. Scooping of the data helps to optimize performance and optimize memory utilization.

APPLICATION OF REAL TIME DISPLAYS

Launch activities can produce a flood of data, which can clog transmission channels. Network traffic is reduced by only transmitting change data information while data fusion allows high sample rates of data to be reduced into manageable data packets. Data that is sampled at high rates, such as a 100 or 1000 samples per second, would be propagated only if the measurement had actually changed. Taking individual measurements and arithmetically or logically processing them into a new value creates fused data. The new value represents a higher level of intelligence than the initial measurement set. Fused data is then distributed to end users in the Launch Control Center rather than distributing numerous individual data elements and having each user perform the same mathematical operation. These approaches to data acquisition greatly reduce the volume of data which needs to be transmitted from the launch pad.

All data is time stamped at a data collection gateway and broadcast to a data distribution processor. Data can then be distributed to the many LCC engineering groups, and can also be accessed from related centers, such as the Mission Control Center in Houston.

NASA is currently considering using a JAVA interface to provide remote centers access to the same Real-time displays used by the launch teams. SL-GMS provides a Java conversion utility, which converts the graphic displays (models) into Java classes which can be run as an applet in a standard web browser. This enhancement will allow related sites to monitor pre-launch activities as they happen while not incurring additional development costs.

Testing and training missions benefit from using the same graphics as the actual launch control systems. Simulated launches can be conducted using historical or simulated launch data. Alarms and other situations are simulated, with the graphics responding in the same manner that they would in actual launch conditions. In training and quality assurance testing, the same graphics used in the actual launch system are connected to the simulation program rather than the actual data distribution engine. Hypothetical scenarios can be created to train launch center personnel and test the effectiveness of new graphical components. A simulation group has mathematical models, which can mimic actual launch scenarios and ensure that launch control systems and engineers are prepared for any eventualities that might occur during an actual launch countdown. Operators can be trained to respond to different problems as they appear on the new operator consoles.

Note 1

After horizontal processing of the Orbiter is complete in the OPF, it is rolled to the Vehicle Assembly Building (VAB) where it is stacked with an empty external fuel tank and two solid rocket boosters. Pyrotechnic bolts are used to mount this vertical assembly to a mobile launch platform (MLP), which is transported via an immense crawler to the launch pad. In order to prepare for flight, the Main Propulsion System (MPS) and the Shuttle's Reactive Control System (RCS) are fueled prior to the start of countdown.

Fuel is piped through transfer lines from storage units to the fixed service unit, where a complex layout of pipes runs to the umbilical connectors and finally into the external tank. New graphic displays allow engineers to better visualize the tanking operations. Operators can glance at a graphic to determine if tanking operations are underway, the level of the fuel tanks and the current status of the count down sequence. Operational status of system hardware (such as pumps and transducers) is also readily apparent. Warnings and alarms are prominently displayed, as graphical elements change appearance when alarm limits are reached. Engineers must continually monitor parameters such as temperature and pressure during

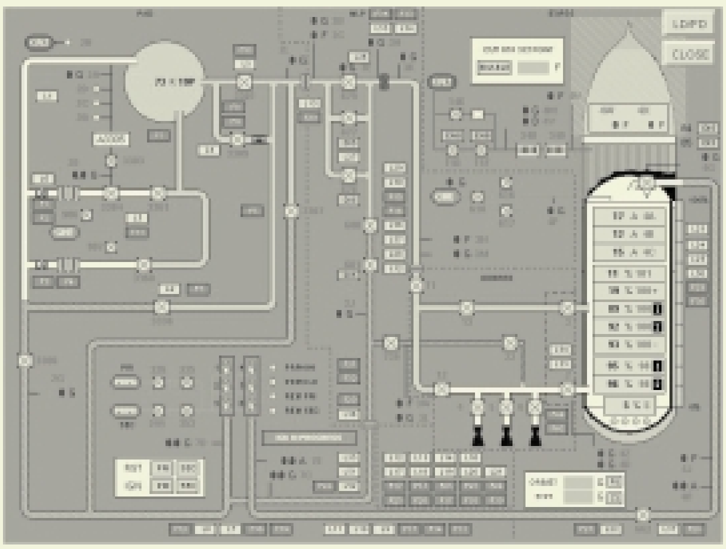

Figure 3.

the tanking operations in order to identify anomalies, which can indicate problems such as hardware failure. Sensors continually monitor air quality and transmit the data back to the LCC. New graphic images display the location of the sensors as well as their live data values. This type of real-time information enhances trouble-shooting procedures. Expediency of error detection is a major benefit of the current system upgrade.

After tanking operations are complete, the astronauts board the Shuttle and begin their preflight checklist. During the final portions of the countdown, the orbiter access arm is retracted, and the gaseous oxygen vent cap is withdrawn from the external tank. From computer consoles in the LCC Engineers monitor the Shuttles onboard systems during the final moments before launch. Time based trend graphs monitor the main engine's health and at T-31 seconds, the Shuttles onboard computers take control of the count down. The main engines start at T-6.6 seconds and (assuming there are no last minutes problems) the solid rocket boosters are ignited and the pyrotechnic bolts, which hold the vehicle to the MLP are blown. The immense thrust from the solid rocket boosters and SSME's lift the vehicle towards the heavens. Once the Shuttle has cleared the tower responsibility for the spacecraft is turned over to Mission Control in Houston.

Note 1

Plaques on the wall of the LCC commemorate the many successful Shuttle launchings conducted at the facility and by upgrading key launch control systems, NASA is preparing to continue this tradition into the next millennium. A reliable Shuttle fleet will help insure that the ISS can be successfully completed and fully utilized. By putting in place new computer hardware and software systems the Kennedy Space Center will continue its leading role in manned space flight well into 21st century. The SL-GMS graphics package will serve as the eyes into critical systems for those charged with the safety of personnel and the successful launching of future Shuttle missions

---

Michael Meagher, Software Engineer, SL Corporation (pictured in Space Shuttle cockpit)

Michael holds a B.S. degree in Mechanical Engineering from the University of California at Berkeley, where he was a member of the Pi Tau Sigma Engineering honor society. He also holds and Economics degree from the University of California at Santa Cruz.

During and immediately following college, Mr. Meagher worked at NASA's Ames Research Center, concentrating on graphics applications for computational fluid dynamics. Mr. Meagher worked on transonic flow solvers as well as graphics programs to display computationaly produced flow data, specializing in Silicon Graphics GL, Fortran and C.

Mr. Meagher joined SL Corporation in 1990, and has