REAL-TIME GRAPHICS FOR THE SPACE STATION FREEDOM CUPOLA, DEVELOPED IN THE SYSTEMS ENGINEERING SIMULATOR

Michael T. Red
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
and
Philip W. Hess
Lockheed Engineering and Sciences Company

ABSTRACT

Among the Lyndon B. Johnson Space Center's responsibilities for Space Station Freedom is the cupola. Attached to the resource node, the cupola is a windowed structure that will serve as the space station's secondary control center. viewing. From the cupola, operations involving the mobile service center and orbital maneuvering vehicle will be conducted.

The Systems Engineering Simulator (SES), located in building 16, activated a real-time man-in-the-loop cupola simulator in November 1987. The SES cupola is an engineering tool with the flexibility to evolve in both hardware and software as the final cupola design matures. Two workstations are simulated with closed-circuit television monitors, rotational and translational hand controllers, programmable display pushbuttons, and graphics display with trackball and keyboard.

The displays and controls of the SES cupola are driven by a Silicon Graphics Integrated Raster Imaging System (IRIS) 4D/70 GT computer. Through the use of an interactive display builder program SES cupola display pages consisting of two dimensional and three dimensional graphics are constructed. These display pages interact with the SES via the IRIS real-time graphics interface. This paper focuses on the real-time graphics interface applications software developed on the IRIS.

LIST OF ACRONYMS AND ABBREVIATIONS

3D	3 dimensional
CCTV	closed-circuit television
CDB	changed data block
CDBRD	CDB read (processor)
CDBWR	CDB write (processor)
CPU	central processing unit
CRT	cathode-ray tube
CVM	current value memory
DLM	display list memory
DLRD	downlink read (processor)
DU	display update (processor)
EXEC	executive (task)
HSD	high speed data
HSDRD	HSD read (processor)
HSDWR	HSD write (processor)
IND	indicator (processor)
INTF	interface (task)
IP	input processor
IP/DU	IP and DU (task)
IRIS	Integrated Raster Imaging System
MSC	mobile service center
OMV	orbital maneuvering vehicle
OTW	out-the-window
PDP	programmable display pushbutton
RISC	reduced instruction set CPU
SES	Systems Engineering Simulator
SW	switch processor
SW/IND	SW and IND (task)
SYSID	system identification

INTRODUCTION

The purpose of simulation is to provide an accurate, economical, and most importantly safe means of testing a product. The product may range from a crewperson's expertise in performing a particular procedure to the procedure itself. Real-time simulation implies that if an event in the real world takes five seconds to transpire, the same simulated event would also take five seconds. Man-in-the-loop simulation places a human in the simulation loop, reacting to the simulation computers. For example, a crewperson initiates a command to a system. The simulation computers receive the command and perform the appropriate response. The crewperson recognizes the response and continues with a new command, completing the simulation loop. Real-time man-in-the-loop simulation provides an individual with the means of performing a task in real time.

The Systems Engineering Simulator (SES) is located in building 16 of the Lyndon B. Johnson Space Center. The SES, depicted in Figure 1, is a real-time man-in-the-loop simulation facility dedicated to providing engineering support for the Space Shuttle and Space Station Programs. SES support covers a wide spectrum ranging from engineering studies to procedures development and crew training.

The SES is composed of a computation facility, scene generation computers, and four crew stations. The computation facility consists of simu-

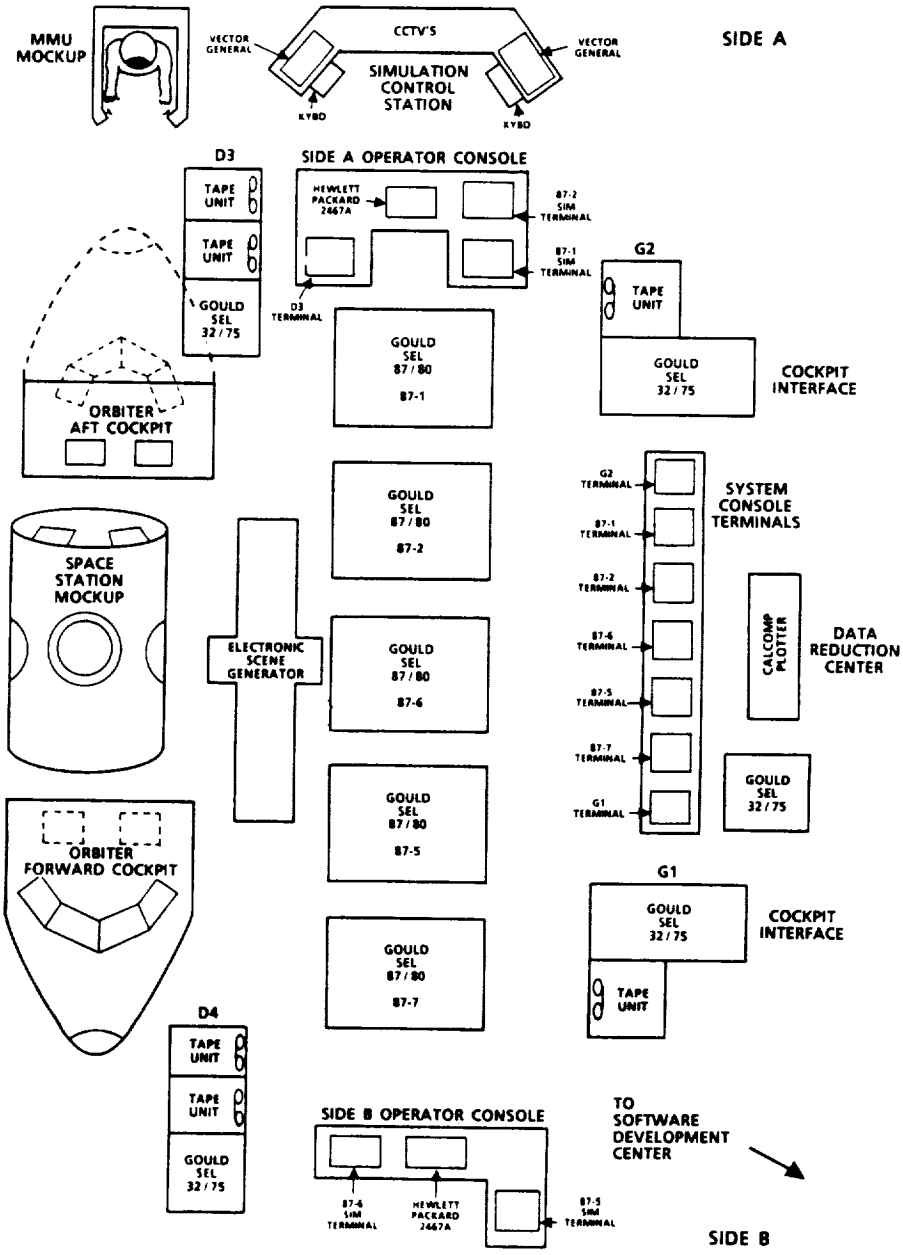

Figure 1
System Engineering Simulator

lation computers, mass storage units, data recording, and development facilities. Three real-time scene generation computers provide a combination of up to eleven out-the-window (OTW) and closed-circuit television (CCTV) views. The four crew stations supported by the SES are the forward shuttle cockpit, aft shuttle cockpit, manned maneuvering unit, and space station cupola.

The space station cupola is the only windowed structure to provide direct line of sight viewing from the space station. In its final phase I configuration the space station will have two cupolas attached to two of the space station nodes. The cupola will serve as the secondary command control station where much of the latter portion of phase I and most of phase II space station assembly will be conducted. Operations of the space station mobile service center (MSC) and orbital maneuvering vehicle (OMV) will also be conducted from the cupola.

The cupola crew station in the SES, referred to as the SES cupola, is designed to be an engineering

tool. With the final configuration not yet established the SES cupola is designed to evolve in both hardware and software configuration. The wooden mockup models half of the cupola with a viewing area for visitors or training personnel in the rear. The crew station portion consists of six OTW views and two side by side crew stations. The next phase of the SES cupola includes many hardware updates driven by McDonnell Douglas, the primary contractor for the space station cupola. This phase III SES cupola, due to be operational in April 1989, will not only include a new physical shell, but also a reconfigured interior and a new OTW optics system. As the design of the space station cupola evolves into a final state, the SES cupola configuration will evolve to match. It is planned that the SES cupola will eventually evolve into a real-time man-in-the-loop simulator with actual flight hardware.

SES CUPOLA HARDWARE

The SES cupola instrumentation and controls consist of several integrated components used to simulate possible flight hardware for two crew stations (Figure 2). The heart of each crew station is a Silicon Graphics Integrated Raster Imaging System (IRIS) 4D/70 GT workstation class computer. The IRIS has a reduced instruction set CPU (RISC) architecture, and therefore processes approximately twelve million instructions per second. This high speed provides quality graphics rendering on top of applications software adequate for a real-time simulation environment.

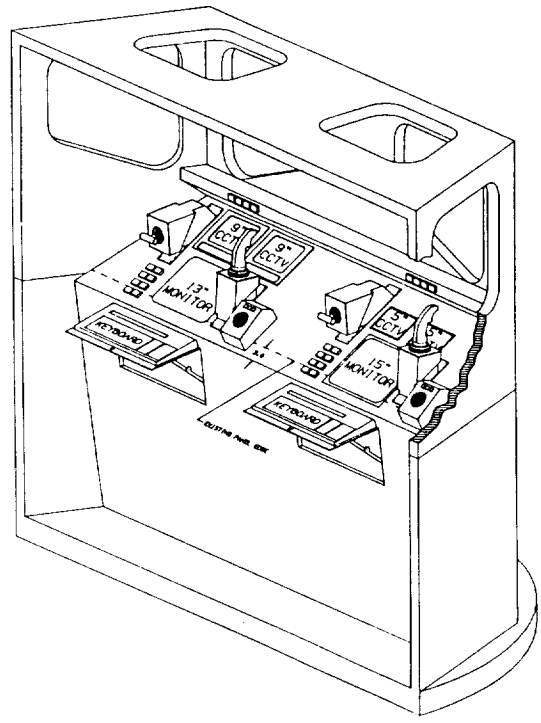

Figure 2
System Engineering Simulator Cupola

The IRIS drives and receives input commands from four user interface components in the crew station. First, the IRIS displays its graphics information on a 1024 raster line by 1280 pixel resolution 15" color monitor. The IRIS receives command input from a keyboard and three button trackball. Finally the IRIS drives the displays and receives command input from three sets of four programmable display pushbuttons (PDP). Rotational and translational hand controllers are currently interfaced to a separate general purpose computer supporting the SES cupola simulator rather than to the IRIS.

A total of three IRIS units are used in the SES. As stated previously, two are used for operations inside the cupola simulator. The third IRIS is used for development. All three IRIS units are connected together through an ethernet interface. One of the IRIS units inside the cupola, referred to as the master IRIS, sends and receives data to and from the SES simulation computers via a high speed data (HSD) interface. The other two IRIS units are referred to as slaves, but only because they receive information from the SES simulation computers via the master IRIS and ethernet. All three IRIS units operate asynchronously from any other computer.

CREW STATION DISPLAYS AND CONTROLS

The displays and controls in the SES cupola provide the user interface to the system a crewperson wishes to access. The focus here is on how information from some probable space station based computer system is displayed to the crewperson as well as how he can input commands to such a system. Therefore, four devices in the crew station will be discussed in detail: the display monitor, three button trackball, keyboard, and PDPs.

As is common with many personal computers and workstations today, the IRIS offers a window management system for flexibility and ease in displaying information. Windowing systems allow the user to display information in a specific portion of the physical CRT screen space. The "window" of information can then be moved from one position on the screen to another. In fact, numerous windows can be displayed on the screen at one time in an overlapping fashion. A user can "pop" a window to the foreground thus allowing all the information in the window to be visible or "push" the window behind all other currently visible windows. A cursor on the screen is usually used to target a specific window for one of the functions mentioned above. The cursor can be moved about the screen by a number of devices including the arrow keys on a keyboard, a mouse unit, and a trackball.

The SES cupola crew station employs a three button trackball instead of a mouse unit to position the cursor on the 15" monitor. Response from astronauts' use of the crew station dictated a preference for the trackball. Restrictions were applied to the IRIS window management system to simplify the operation of the crew station. For example, windows can be popped to the foreground but cannot be pushed to the background, and windows cannot be

reshaped. Furthermore, specific functions were tied to the three buttons on the trackball. The right button only pops windows. The middle button only moves windows. The left button is used only to select functions on the screen such as a switch. By limiting the complexity of the window management system through dedicated trackball buttons, the crewperson interfaces with an extremely user friendly system with little chance of error on the user's part.

The IRIS real-time applications software recognizes three types of display windows: data, banner, and pop-up windows. Data windows are by far the most common. They can be moved with the middle trackball button and popped with the right trackball button. As the name implies data windows display data, but they can also be used to call up new windows as well as receive inputs. Pop-up windows cannot be moved or popped to the foreground. They are designed to emulate pull down menus and thus are used only to call up new windows. When a pop-up window is called up, the next depression of the left trackball button is expected to be inside the pop-up window. Otherwise, the window is deleted. The banner window is the most unique display window. The banner window covers the entire screen and contains simulation status and time information as well as the ability to call up other windows. It cannot be moved, deleted, or popped to the foreground. The banner window is brought up when the IRIS real-time applications software is initialized and remains present during the entire simulation run with all other display superimposed.

The keyboard in the current SES cupola crew station has a very limited function. During simulation operations there is a keyboard display type available on some display windows. This display type requires keyboard input from the crewperson in the form of a floating point number. McDonnell Douglas, as a future user of the SES cupola, has requested increased use of the keyboard.

One of the thrusts behind the design of the space station cupola is a reduction in the number of hardware switches due to the lack of available space. The use of twelve PDPs per crew station in the SES cupola is one method of reaching this design goal. As the name implies PDPs can be programmed for numerous functions at different points in time. For example a specific PDP may be programmed to pan a CCTV camera to the right when depressed. Later the same PDP may be programmed to trigger the snares in the MSC end effector to capture a target. In this way the total number of hardware switches in the cupola can be significantly reduced. Currently, in the SES cupola PDPs are used to pan, tilt, and zoom CCTV cameras, as well as control several MSC functions including: turning off the master alarm, turning MSC brakes on or off, driving individual MSC joints positive or negative, and triggering the capture or release of a target.

SES CUPOLA DISPLAY AND CONTROL MONITOR

One of the primary concepts in development of the IRIS real-time applications software for the SES cupola was flexibility. Past experience had proven that hard coded display windows were difficult to modify and maintain. Because the SES cupola is an engineering simulator, the ability to modify display windows with minimal turnaround time is very important to many potential customers. It is not unreasonable that they may wish to try several different display window layouts. The IRIS real-time applications software must be flexible enough to change the display window layout as quickly as possible. Therefore, all display windows, regardless of the type (data, pop-up, or banner), are read from display data files.

The concept of the display file (Figure 3) is very straight-forward. Each display file is constructed with an off-line display builder program and contains all the information needed to produce a display window in the real-time applications software. The information in the display file is organized into units referred to as display types. Therefore, when a window is called up by the real-time system, such as the banner window upon initialization, a specific display file is read, and the display types in that file are used to draw the window during operations.

There is a finite number of defined display types that are recognizable to both the display builder and the real-time applications software. However, one of the major advantages to this method of constructing display windows is that new display types can be added with minimal impact. Once a display type is defined, construction and modification of display files becomes almost a trivial process due to the user friendly nature of the display builder. Display types in general contain the following information: an opcode to designate the type, the number of words used to define the display type, a system identification number (SYSID) used for variable data related to the display type, and (x,y) coordinates to position the display type in the display window. Beyond this preliminary data, display type information becomes more specific to the actual display type. A list

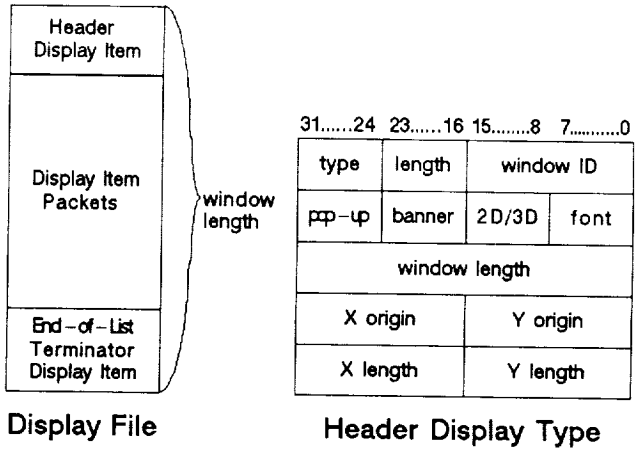

Figure 3
Display File and Header Display Type

of currently available display types is provided in Table I and Table II, and a more detailed description of each display type can be found in Appendix A.

Table I
2 Dimensional Display Types

Header Entry	End of List Terminator
Integer	Real
Long Float	Hexadecimal
ASCII Message	Static Text
Page Call	Keyboard Input
Delete	Default
Circular Gauge	Meter Bar
Indicator	Switch
Momentary Switch	Line
Rectangle	Circle
Polygon	Dynamic Position Indicator

Table II
3 Dimensional Display Types

Header Entry	End of List Terminator
Begin Coordinate Frame	End Coordinate Frame
Rotation	Translation
Scale	Box
Cylinder	Sphere
Line	

Another concern related to display windows is color. Programs prior to space station have implemented monochrome display systems and have not had to necessarily deal with the potential excessive use of color in a display system. Astronaut response to displays in the SES cupola has indicated that a wide variety in color is distracting. The number of available colors in the SES cupola has been limited to sixteen, including white and black. The complete list of available colors is provided in Table III. An attempt has been made to reduce the amount of color actually used in display windows. For instance, switches in the off position and indicators in the false state are colored gray by convention. In general green is used to indicate an active or true state, yellow indicates caution, and red indicates a warning. The other colors are used discriminatingly always attempting to reduce the amount of color on the display.

Most displays use only 2 dimensional display types. However, the graphics capability of the IRIS supports 3 dimensional (3D) graphics. While the cupola is a structure with several windows, a large percentage of the potential field of view is obstructed. CCTV cameras help, but it is very simple to lose your orientation. Radar and telemetry data from vehicles in the proximity of the space station could be used to develop a 3D situation display. The 3D situation display simulates this capability. It contains 3D wire frame drawings of the space station with MSC and vehicles in proximity of the space station (e.g., orbiter and OMV) in the correct orientation and position relative to each other. The display can be rotated, translated, or zoomed, giving the crewperson an excellent omniscient view of vehicles in proximity of the space station.

Table III
SES Cupola Graphics Monitor Colors

Black	Bright Red
Bright Green	Bright Yellow
Dark Blue	Magenta
Cyan	White
Dark Red	Dark Green
Dark Yellow	Blue
Orange	Purple
Gray	Blinking Red

REAL-TIME SYSTEM DATABASE

The SES cupola applications software utilizes an indexed shared memory concept that allows for rapid modification of shared memory (Figure 4). The allocated shared memory, called current value memory (CVM) is divided into two sections: the pointer section and the data section. The pointer section is in the lower address portion of CVM. As the name implies the pointer section contains pointers to the higher address portion of CVM or data section. The offset from the CVM base address corresponds to the SYSID of the variable. Therefore, the length of the pointer section is defined by the largest SYSID. The data section of shared memory contains data packets described below.

An off-line database program is used to maintain the SES cupola CVM. The Informix relational database program is used to keep track of all SYSID variables. Information such as the variable name and description, where it resides in the uplink or downlink buffer, as well as the variable type and initial value are kept in the database.

Applications programs were written to access the Informix database and extract information needed to build three data files: the memory image file, the uplink parameter file, and the downlink parameter file. The memory image file is a replica of the SES cupola CVM during operations (Figure 5). It contains all of the data packets that will reside in CVM. Each packet has the variable type (ie. floating point, double precision floating point, signed or unsigned 32-bit integer, 16-bit integer, bit, or character string), the length in words, the SYSID of the variable, and the current value of the variable. The uplink and downlink parameter files represent a mapping from CVM to the buffer of data sent to (uplink) and from

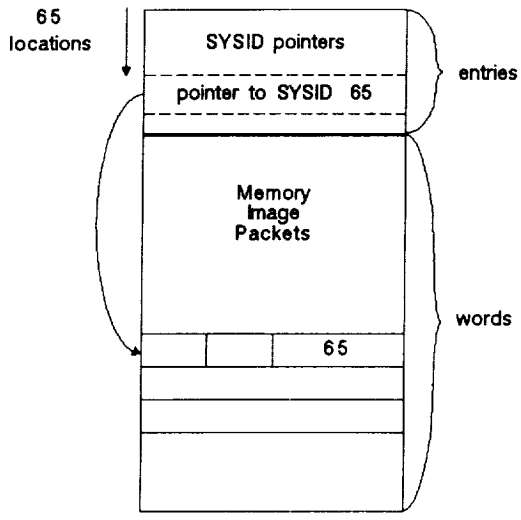

Figure 4
Current Value Memory Format

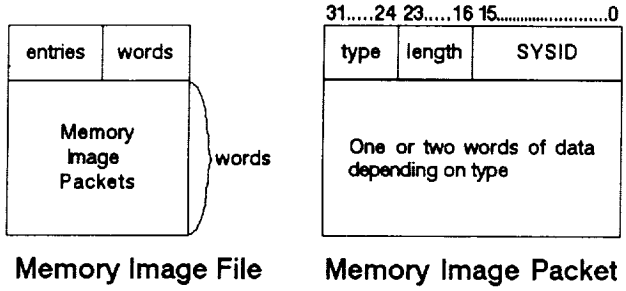

Figure 5
Database Memory Image File

(downlink) the simulation computers by the master IRIS via the HSD interface. The parameter file is ordered by buffer word first and if necessary by bit location second. Each entry in the parameter file contains the SYSID of the variable, word location in the buffer, and start bit.

Upon initialization of the real-time applications software, the executive task reads the memory image file, extracting the size of CVM, and dynamically allocates enough shared memory for CVM. As the data packets are read from the memory image file and placed into the data section of CVM, the pointer in the pointer section is resolved for the appropriate SYSID. This process continues until the memory image file is completely read.

There are some important advantages to this concept of accessing and maintaining data. First, the database is maintained off-line, and is always current to the real-time applications software because it is always read in each time the SES cupola is initialized. Second, all SYSIDs within a particular range do not have to used. In other words, there can be "holes" in the database. Third, holes in the database effect only the pointer section of CVM; the data section is always compressed with no wasted memory. Fourth, the database is used to generate reports that document the uplink and downlink buffer definitions; list SYSID variables associated with particular subsystems; list SYSID variables not in use; list the database sorted by variable type, name, SYSID, subsystem, and other criteria.

SES CUPOLA REAL-TIME SYSTEM SOFTWARE

The SES cupola real-time applications software for the IRIS was developed in-house by NASA and support contractor personnel (Figure 6). Several guidelines were adhered to in development of the real-time system to facilitate maintenance. First, the real-time system would be machine independent. Only one version of the real-time system would exist and be run on both the master and slave IRIS units. Second, the real-time system would be broken down into major functions. These major functions would reside in separate tasks so that if changes were made to a specific task and the real-time system failed, then that task would be suspect. Third, an executive task would be used to initiate and schedule the real-time system. Along with the executive (EXEC) task the real-time system is made up of the input processor and display update (IP/DU) task, the switch processor and indicator processor (SW/IND) task, PDP task, HSD task, and interface (INTF) task.

The EXEC task is the heart of the real-time system. It is responsible for allocating and initializing CVM, the changed data block (CDB) which will be detailed later, and executive shared memory which contains variables needed by other tasks in the real-time system. EXEC is also responsible for initiating the other five tasks in the real-time system as well as establishing communications between itself and the other tasks. Finally, EXEC is responsible for scheduling the other tasks in the real-time system. Considerable attention was given to the problem of homogeneous data in CVM. The order of scheduling shown in Figure 7 insures that by the time display related processing is begun in the SW/IND, CVM is updated.

Each of the five subordinate tasks contain a task executive and the processes that actually perform the function of the task. The task executive (not to be confused with EXEC) is essentially generic from subordinate task to subordinate task. Its purpose is to initialize the task in terms of access to CVM, executive shared memory, and CDB if necessary. The task executive also allows its processes to initialize if necessary. Finally, the task executive completes establishment of communications with EXEC. Once the task executive has finished initialization, it enters an infinite run loop and is put to sleep until EXEC signals it to go.

The IP/DU task contains two separate processors: the input processor (IP) and the display update

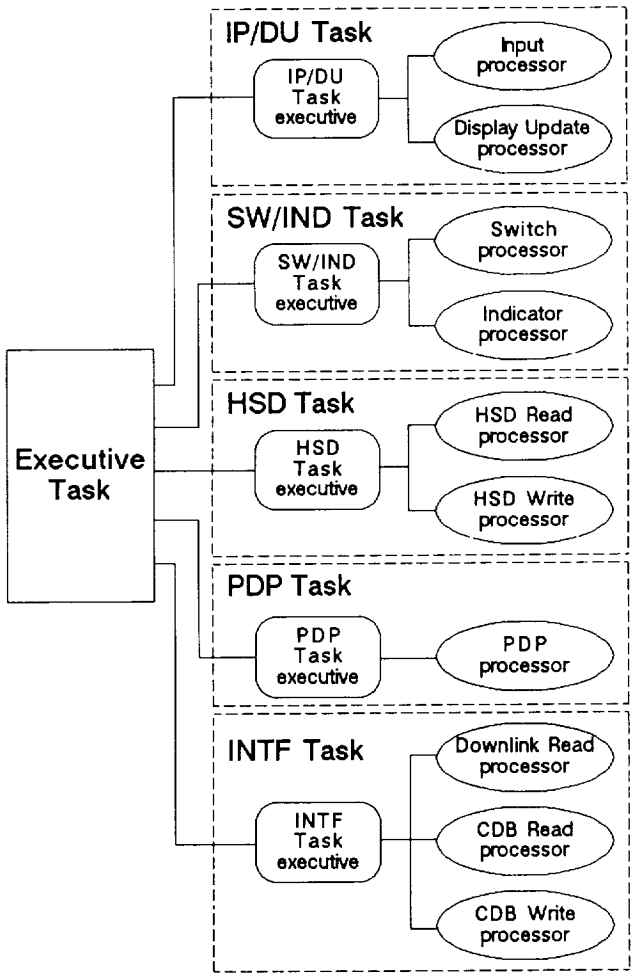

Figure 6
SES Cupola Real time Applications Software

processor (DU). The IP/DU task executive uses information in executive shared memory to determine which process to execute when it is signaled to go by EXEC.

The IP, as the name implies, processes input information from the keyboard and three button trackball. Upon initialization the IP reads the display file for the banner window and places the display information in dynamically allocated memory, called display list memory (DLM). The IP resolves overlapping display windows, as well as allocation and deletion of display windows. When a position in a display window is selected with the left button of the trackball, the IP determines the cursor position and compares it to display item positions in the currently allocated DLM. Once the proper display item is found, the IP executes the appropriate function based on the type of the display item. For example, if a switch was selected, the appropriate switch SYSID is set true, or if a page call was selected the appropriate display file is read and placed in DLM. When a display window is selected with the middle trackball button, the IP is responsible for moving that window. Finally, when a display window is selected with the right trackball button, the IP is responsible for popping that window to the foreground. Currently, the IP processes keyboard input only if the keyboard display type is selected with the left trackball button.

The DU is responsible for update of all display windows. The DU begins a trace of DLM at the appropriate starting point for a particular display window. As it encounters each display type, the DU executes the graphics commands to draw that display type. The DU must also perform some calculations to correctly draw the display type. For example, the gauge display type has a needle that must be positioned correctly based on the limits of the gauge and the current value of the gauge in CVM. The DU must perform the appropriate calculations to correctly position the gauge needle. In this manner the DU uses CVM to correctly display gauges, meter bars, switches, indicators, and any other display type that changes based on its associated SYSID value in CVM.

The SW/IND task contains two separate processors: the switch processor (SW) and the indicator processor (IND). The SW/IND task executive uses information in executive shared memory to determine which process to execute when it is signaled to go by EXEC. SW/IND is the only task in the real-time system that must be hard coded with SES cupola specific functions.

The SW is responsible for resolving switch selection in the SES cupola. It is through the SW that mechanical devices such as rotary switches are duplicated in software. For example, a bank of switches on a display window may have the implied function that no two switches may be selected at any one time (a rotary switch). The SW resolves which switch has been selected and deselects all of the other switches in the bank.

The IND works closely with the SW to perform hardware functions in software. Like the SW, the IND must resolve some banks of indicators where only one indicator in the bank may be active at one time. However, the IND also deciphers the time data from the simulation computers to correctly display mission elapsed time and Greenwich Mean time.

The PDP task is responsible for processing input from the PDPs and updating the PDP displays. Upon initialization the PDP task reads a data file designed to establish the PDP configuration. As with the display files, it is the PDP data file that defines the PDP configuration for the SES cupola; the real-time PDP software is generic and simply responds to the data file. The PDP data file defines which switches are momentary (active only when depressed) and which change state on each depression. The data file also defines the PDP tree structure. For example, one PDP may be used to reconfigure an entire bank of PDPs.

The HSD task is responsible for communications

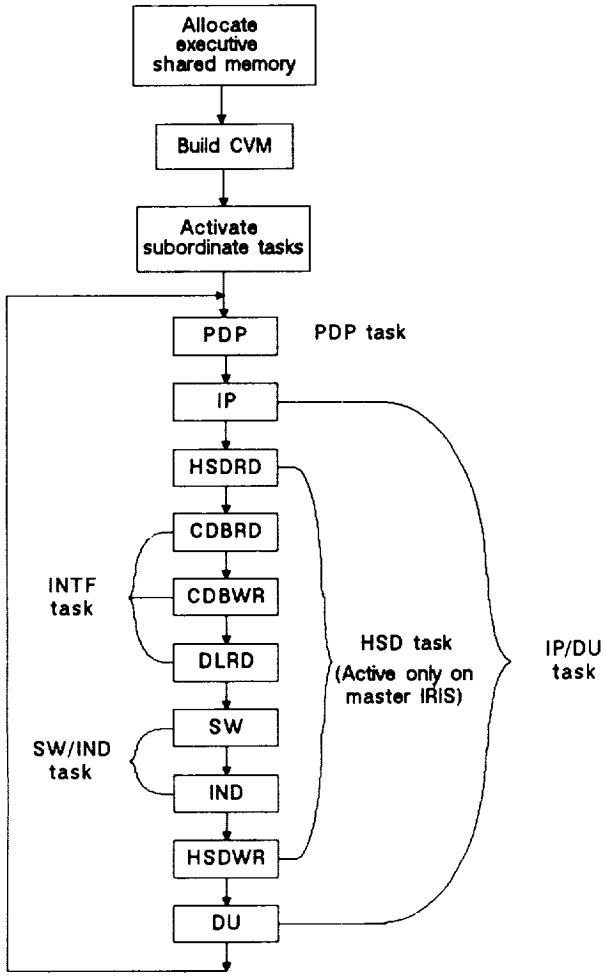

Figure 7
Executive Task Flow
and Process Scheduling

with the HSD interface. This task is active only on the master IRIS and has two separate processors: the HSD read processor (HSDRD) and the HSD write processor (HSDWR). The HSD task executive uses information in executive shared memory to determine which process to execute when it is signaled to go by EXEC.

As stated previously the IRIS computers operate asynchronously from each other and all other computers. Therefore, information from the simulation computers is used only when the master IRIS asks for it. Timing data indicates that in general the master IRIS requests information more often than the simulation computers are prepared to offer it. The HSDRD retrieves a buffer of downlinked information from the simulation computers if available. If a buffer of data is received then it is immediately broadcasted to all IRIS units on the ethernet so that each unit may properly update CVM.

Upon initialization the HSDWR reads into dynamically allocated memory the uplink parameter file built from the Informix database explained earlier. The HSDWR uses the uplink parameter table to map information from the master IRIS CVM into a data buffer that the simulation computers will understand. This data buffer is then sent to the simulation computers. It is important to note that only the master IRIS CVM is used as the source to build the uplink buffer. The INTF task is responsible for making the CVM on each IRIS machine identical.

The INTF task works closely with the HSD task in the area of communication. However, while the HSD task is most concerned with the HSD interface, the INTF task is involved solely with the ethernet interface. The INTF task contains three separate processors: the downlink read processor (DLRD), the CDB write processor (CDBWR), and the CDB read processor (CDBRD). The INTF task executive uses information in executive shared memory to determine which process to execute when it is signaled to go by EXEC.

Upon initialization the DLRD reads into dynamically allocated memory the downlink parameter file built from the Informix database explained earlier. When the HSDRD routine broadcasts the buffer of downlinked data from the simulation computers, the DLRD retrieves that buffer of data. The DLRD then uses the downlink parameter table to map the data from the downlink buffer into CVM.

The CDB is the real-time systems method of porting changes made to one IRIS' CVM to all other IRIS units on the ethernet. For example, during SES cupola operations, a crew person at one station selects a switch. The crew person at the other station expects his display to reflect that switch selection. This is accomplished through the CDB. The CDBWR transmits the CDB across the ethernet to other IRIS units, and the CDBRD receives the CDB from other IRIS units and updates CVM.

FUTURE SES CUPOLA CONSIDERATIONS

As stated earlier, one of the objectives of the SES cupola is to provide to the engineering community a tool for development of the space station cupola. As the hardware design of the cupola changes, the SES cupola hardware will undergo incremental changes. Also a dome visual system is planned for the SES cupola in the future adding another dimension of realism to man-in-the-loop simulation.

By its very nature, software is more flexible than hardware. As this paper has demonstrated, the SES cupola real-time system was designed for flexibility and change. The offline software tools (display builder and relational database) are integral parts of the real-time systems built in flexibility.

With these concepts in mind the future of the SES cupola is bright. Currently OMV simulation in the SES is undergoing validation. The SES cupola crew station will be used as both a ground based con-

trol station for OMV operations and a space based control station. Likewise, the MSC is currently being implemented in the SES. Many of the MSC operations will be developed and analyzed from the SES cupola. Beginning in April 1989, McDonnell Douglas will utilize the SES cupola for displays and controls development. A number of requests concerning software modification have been made by McDonnell Douglas, and those changes are currently in work.

It is through its flexibility and ability to adapt to the needs of the sponsor that the SES in general becomes an excellent engineering tool. The SES has been directed to be the primary real-time man-in-the-loop engineering simulation facility for support of the Space Station Program. The SES cupola is a precise and visible attempt to meet that directive.

REFERENCES

1. St. John, Robert H.; Moorman, Gerard J.; and Brown, Blaine W., "Real-time Simulation for Space Stations", Proceedings of the IEEE, Vol. 75, No. 3, March, 1987, pp. 383-398.

2. "Systems Engineering Simulator (SES) Laboratory Description Document: Simulator Foundation", Vol. 1, LEMSCO-23181, December, 1987.

3. "Systems Engineering Simulator (SES) Laboratory Description Document: On-Orbit Element Simulator", Vol. 3, LEMSCO-23183, April, 1988.

Appendix A

SES Cupola Display Type Descriptions

The IRIS 4D supports the SES Cupola through display windows on the multi-purpose applications console (MPAC) as well as keyboard and trackball (or mouse) input. The display windows respond to the SES via a data buffer through M1. Variables downlinked from the SES to the IRIS are displayed in various formats depending on the display type used. Likewise, inputs from the crewperson are interpreted by the IRIS and uplinked to the SES.

Everything shown on a display window is a display type. All variable data as well as static data is defined in terms of display types. So a display type is simply a functional unit on the display. Variable data display types are used to display data from the SES. These display types tag a unique number termed a system identification (sysid) to their variable so that the real-time SES Cupola software can keep track of variables passed to and from the SES as well as internal variables. The following is a description of display types required for the SES Cupola MPAC.

2D Header Entry - The header entry is at the beginning of every display file that describes a MPAC display window. It contains data that describes the window as a popup window, banner window, or data window. A banner window is a unique window in the real-time system that covers the entire display screen and cannot be popped to the foreground, moved, or deleted. A popup window usually contains a number of page calls (described later) to bring up data windows of related data. Data windows present data to the crewperson. The font used to display text is determined by the font flag. The header entry also contains the length of the window data display file in bytes, the X and Y coordinates of the window origin, and the X and Y length of the window in pixels.

3D Header Entry - The 3D header entry is at the beginning of every display file that describes a MPAC 3D display window. All 3D display windows are data windows. The font used to display text is determined by the font flag. The 3D header entry contains the length of the window data display file in bytes, the X and Y coordinates of the window origin, and the X and Y length of the window in pixels. The distance of the eyepoint from the origin, near clip plane, and far clip plane distances are specified. Finally, the perspective angle and z-buffer flag are specified.

Begin Coordinate Frame - The begin coordinate frame display type is associated with 3D display windows. This display type causes all subsequent 3D objects to be drawn relative to the relocated local origin as specified by the six data variables. Six sysids tag data variables for X, Y, and Z position and X, Y, and Z rotation. A display type name is also specified.

End Coordinate Frame - The end coordinate frame display type is associated with 3D display windows. This display type cancels the coordinate frame display type and the local origin is returned to the previous global origin. All coordinate frames must be terminated by an end coordinate frame. A display type name is specified.

Rotation - The rotation display type is associated with 3D display windows. This display type allows the user to rotate the eyepoint about the global origin. The X, Y, and Z rotations are specified deltas. A sysid tags the data variable that is used to determine if the eyepoint is to be rotated. A display type name is also specified.

Translation - The translation display type is associated with 3D display windows. This display type allows the user to translate the eyepoint along the X, Y, and Z axis a specified distance. The X, Y, and Z translations are specified deltas. A sysid tags the data variable that is used to determine if the eyepoint is to be translated. A display type name is also specified.

Scale - The scale display type is associated with 3D display windows. This display type allows the user to scale a pre-defined amount about the origin along any or all axes. The X, Y, and Z scale factors are specified deltas. A sysid tags the data variable that is used to determine if the display is to be scaled. A display type name is also specified.

End of List Terminator - The end of list terminator is at the end of every display

file that describes a SES Cupola MPAC display window. The sole purpose of the end of list terminator is to flag the end of the display list.

Integer - The integer display type is used to display integer data on the display window. A sysid tags the data variable. The X and Y window coordinates of the display type and field width are specified. A yellow (warning) threshold and red (critical) threshold are available. The integer is also described as increasing or decreasing so that the thresholds reside at the upper or lower end of the expected range. Normally the data is displayed in white. If the value crosses the warning threshold the data is displayed in yellow. Likewise, if the data crosses the critical threshold the data is displayed in red. Thresholds are optional and both thresholds do not have to be used.

Real - The real display type is used to display floating point data on the display window. A sysid tags the data variable. The X and Y window coordinates of the display type, total field width, and number of digits to the right of the decimal point are specified. A yellow (warning) threshold and red (critical) threshold are available. The real number is also described as increasing or decreasing so that the thresholds reside at the upper or lower end of the expected range. Normally the data is displayed in white. If the value crosses the warning threshold the data is displayed in yellow. Likewise, if the data crosses the critical threshold the data is displayed in red. Thresholds are optional and both thresholds do not have to be used.

Long Float - The long float display type is used to display double precision floating point data on the display window. A sysid tags the data variable. The X and Y window coordinates of the display type, total field width, and number of digits to the right of the decimal point are specified. A yellow (warning) threshold and red (critical) threshold are available. The double precision number is also described as increasing or decreasing so that the thresholds reside at the upper or lower end of the expected range. Normally the data is displayed in white. If the value crosses the warning threshold the data is displayed in yellow. Likewise, if the data crosses the critical threshold the data is displayed in red. Thresholds are optional and both thresholds do not have to be used.

Hexidecimal - The hexidecimal display type is used to display a 32-bit word in memory on the display window in the form of a hexidecimal number. A sysid tags the data variable. The X and Y window coordinates of the display type and color are specified.

Ascii Message - The ascii message display type is used to display ascii text that is variable such as error messages on the display window. A sysid tags the ascii data. The X and Y coordinates of the display type as well as color are specified.

Static Text - The static text display type is used to display ascii text that is static such as labels on the display window. The X and Y coordinates of the display type as well as color are specified. The character string is a maximum of 8 characters in length.

Page Call - The page call display type is used to "call up" new display windows for the MPAC. The display type bounds (left X, right X, bottom Y, top Y), color, and text are specified. A filename is specified to indicate the display file to be read.

Keyboard Input - The keyboard display type allows user input from the keyboard. The display type bounds (left X, right X, bottom Y, top Y), color, and text are specified. A filename is specified to indicate the display file to be read.

Delete - The delete display type allows the user to delete a display window in real time. The display type bounds (left X, right X, bottom Y, top Y) are specified.

Default - The default display type allows the user to save a screen configuration of several display windows and recall that particular configuration at some later time. The display type bounds (left X, right X, bottom Y, top Y) as well as the middle Y position and color are specified. Text labels for the default and save default portions of the display type are specified. Finally, the name of the save file is specified.

Dynamic Position Indicator - The dynamic position indicator is a cursor on a bar. The position of the cursor is determined by the associated data variable and the specified upper and lower limits of the indicator. A sysid tags the data variable. Another sysid tags a visibility flag which is used to determine if this display type is drawn. The cursor may take the following forms: empty square with "X", filled square, empty circle with cross-hair, filled circle, empty triangle, filled triangle, caret, or cross-hair. The bar may be vertical or horizontal. The display type bounds (left X, right X, bottom Y, top Y), bar color, and cursor color are specified. The upper and lower

limits are specified.

Circular Gauge - The circular gauge display type is used to display floating point data in the form of a gauge on the display window. A sysid tags the data variable. The X and Y window coordinates of the display type, total field width, and number of digits to the right of the decimal point are specified. The upper and lower limits of the gauge are specified. A yellow (warning) threshold and red (critical) threshold are available. The gauge is also described as increasing or decreasing so that the thresholds reside at the upper or lower limits of the gauge. Normally the data is displayed in white. If the value crosses the warning threshold the data is displayed in yellow. Likewise, if the data crosses the critical threshold the data is displayed in red. Thresholds are optional and both thresholds do not have to be used.

Meter Bar - The meter bar display type is used to display floating point data in the form of a meter bar on the display window. A sysid tags the data variable. The display type bounds (left X, right X, bottom Y, top Y), total field width, and number of digits to the right of the decimal point are specified. The upper and lower limits of the meter bar are specified. A yellow (warning) threshold and red (critical) threshold are available. The meter bar is defined as horizontal or vertical and with or without threshold and limit labels. The meter bar is also described as increasing or decreasing so that the thresholds reside at the upper or lower limits of the meter bar. Normally the data is displayed in white. If the value crosses the warning threshold the data is displayed in yellow. Likewise, if the data crosses the critical threshold the data is displayed in red. Thresholds are optional and both thresholds do not have to be used.

Indicator and Rounded Indicator - The indicator display type represents a mechanical light indicator on the display window. A sysid tags the data variable. The display type bounds (left X, right X, bottom Y, top Y) are specified. Also, the "true" state text, text color, and background as well as the "false" state text, text color, and background are specified. If the variable associated with the indicator is 0 (False) the "false" text, text color, and background are displayed. Any other value is considered true, and the "true" text, text color, and background are displayed. The rounded indicator display type has rounded ends.

Switch - The switch display type represents a mechanical 2-way toggle switch on the display window and is drawn to create the illusion of a 3-D push button. An additional feature of the switch display type is that an indicator can be incorporated into the switch. A sysid tags the switch data variable, and another sysid tags the indicator data variable. The display type bounds (left X, right X, bottom Y, top Y) are specified. Also, the "true" state text, text color, and background as well as the "false" state text, text color, and background are specified. A transition color is specified if the indicator option is used. The truth table below indicates the state of the switch based on the value of the data variables. Note that if the switch sysid (ss) and indicator sysid (is) are identical the switch acts as a 2-way toggle. If the two sysids are different then the switch has a transition state.

switch sysid = indicator sysid

ss	is	position	color
0	0	up	false
1	1	down	true

switch sysid <> indicator sysid

ss	is	position	color
0	0	up	false
0	1	up	true
1	0	down	transition
1	1	down	true

Momentary Switch - The momentary switch display type represents a mechanical 2-way toggle momentary switch on the display window and is drawn to create the illusion of a 3-D push button. An additional feature of the display type is that an indicator can be incorporated into the momentary switch. A sysid tags the switch data variable, and another sysid tags the indicator data variable. The display type bounds (left X, right X, bottom Y, top Y) are specified. Also, the "true" state text, text color, and background as well as the "false" state text, text color, and background are specified. A transition color is specified if the indicator option is used. The truth table below indicates the state of the momentary based on the value of the data variables. Note that if the switch sysid (ss) and indicator sysid (is) are identical the momentary acts as a 2-way toggle. If the two sysids are different then the momentary has a

switch sysid = indicator sysid

ss	is	position	color
0	0	up	false
1	1	down	true

switch sysid <> indicator sysid

ss	is	position	color
0	0	up	false
0	1	up	true
1	0	down	transition
1	1	down	true

Line - The line display type draws a line on the display window. The starting X and Y coordinates, ending X and Y coordinates, and color are specified.

Rectangle - The rectangle display type draws a rectangle on the display window. The display type bounds (left X, right X, bottom Y, top Y) and color are specified. The rectangle can be filled or empty.

Circle - The circle display type draws a circle on the display window. The display type X and Y coordinates, radius, and color are specified. The circle can be filled or empty.

Polygon - The polygon display type draws a polygon on the display window. The polygon can have up to and including 10 vertices. The number of vertices, all (X, Y) coordinate pairs, and color are specified. The polygon can be filled or empty.

3D Line - The 3D line display type draws a line in three dimensional space and is used on 3D display windows. Starting X, Y, Z and ending X, Y, Z coordinates are specified. A display type name and color are specified also.

Box - The box display type draws a box on the 3D display window. The center X, Y, and Z coordinates; height and width on the -X and +X ends of the box; and length are specified. An offset along the Y or Z axis may be specified to shift the front face of the box. Rotations about the three axes may be specified to orient the box. A display type name and color are specified also.

Cylinder - The cylinder display type draws a cylinder on the 3D display window. The X, Y, and Z coordinates; diameter at the -X and +X ends of the cylinder; and length are specified. The number of sides and angle of rotation (full cylinder = 360, half cylinder = 180, etc.) are specified. Rotations about the three axes may be specified to orient the cylinder. A display type name and color are specified also.

Sphere - The sphere display type draws a sphere on the 3D display window. The X, Y, and Z coordinates of the center, radius, and number of sides are specified. A display type name and color are specified also.