The Space Congress® Proceedings | 1988 (25th) Heritage - Dedication - Vision

Apr 1st, 8:00 AM

The Generic Checkout System Approach to Ground Checkout Systems

Peter C. Simons
McDonnell Douglas Astronautics Co.

The Generic Checkout System Approach to Ground Checkout Systems A Paper for the Twenty-Fifth Space Congress

Peter C. Simons

McDonnell Douglas Astronautics Co.

ABSTRACT

With the advent of the Space Station, Heavy Lift Launch Vehicle and other projects, NASA has been prompted to critique major ground checkout systems around KSC. This critique is being used as a basis for the development of a central set of functions which are common to all checkout operations throughout the program. A prototyping effort was started over a year ago to implement those central functions, this effort was called the Generic Checkout System(GCS) which, over the past year, has grown into a working model 1 for ground checkout systems. At the Twenty-Third Space Congress a paper was presented which outlined the rudimentary operations of the GCS. Since that time GCS has evolved into a state of the art checkout system which demonstrates flexibility and ease of use. The GCS system has been chosen as the architecture which will support the Partial Paylod Checkout Unit(PPCU), a new system to be installed in early 1990. The development of the GCS system was meant to also address several problems inherent in current checkout systems: lack of flexibility, poor user interfaces and the abscence of an upgrade path from obsolete hardware. The GCS seeks to solve these problems in ways which utilize high technology advances in computer hardware and software. These advances include the use of commercial UNIX operating system based computers which offer vendor independence and portability of software, the use of state of the art user interfaces offering high resolution graphics, mouse interfaces and the ability to create displays interactively without the need to generate code to drive them. The use of other high tech products is also apparent in the GCS such as the support for Artificial intelligence, relational data base technologies, ADA programming language, parallel processing, RISC technology architectures, optical storage media, Local Area Network Connectivity, commercial graphics packages, INMOS transputers and the latest microprocessor technologies. This paper will attempt to explore some of the facets of the GCS prototyping and development effort and mention the future plans for the architecture which has been developed.

1. INTRODUCTION

The use of complex, high technology computer systems for ground checkout at KSC has long been a tradition. Automated, computer based ground checkout systems are used throughout the payload and orbiter processing flow. As a whole, however, these systems are aging and becoming more difficult to maintain and operate. With the onset of new programs in space such as the Space Station, Heavy Lift Launch Vehicle, Lunar Base, etc. a hard look was taken at the existing ground checkout systems and their ability to handle these complex and long term mission requirements. Investigation of existing systems made it apparent that the checkout systems existing at the Kennedy Space Center were not adequate to do the job. Many of the systems utilized outdated systems which were no longer manufactured or supported by the manufacturer. Maintenance had become a costly and time consuming endeavor. It was also evident that the many systems in existence performed similar functions, that is, the basic functions for ground checkout operations existed to some extent in all ground checkout systems. This prompted NASA to undertake the creation of the GCS(Generic Checkout System). This system would be used as a prototype hardware and software testbed for the creation of a core set of checkout functions which could then be applied to the creation of several checkout systems. The Generic Checkout System would possess all those basic functions which were common throughout the program. The GCS would, in addition, build on the mistakes and successes of the past to provide a flexible, upgradeable and highly usable system for ground checkout. This paper will address the progress of GCS system development since its inception in 1985 and forecast the future of the technologies developed in the GCS project.

2. PROJECT HISTORY

The GCS development effort started in late 1985 as a study for the upgrade of the existing Control Checkout and Monitor Subsystem of the Space Shuttle Launch Processing System. It was decided early that a multi-phased approach to the study effort would be applied. Each phase of the implementation would add functionality to the previous phase until a final operational Checkout System was completed. The initial phase, PHASE I, of the GCS development was geared toward the Launch Support arena. A system which simulated the existing Control Checkout and Monitor user interface was developed using high resolution graphics. In addition several aspects of portability were introduced as well as the porting of the Ground Operation Aerospace Language to a UNIX environment. The Ground Operation Aerospace Language(GOAL) is a high level, procedural language used in CCMS and other systems for the automation of checkout procedures for launch and ground checkout operations.

PHASE II of the GCS development effort was used to implement all major subsystems and attempt to benchmark the usability of the GCS architecture. These benchmarks included Network throughput analysis, analysis of real time extensions to UNIX System V, Portable Graphics analysis and actual control of a test article. In addition new hardware technologies were inserted into the system to test software portability. The PHASE II system is complete and currently supporting demonstrations at the Kennedy Space Center.

PHASE III GCS is currently in the design phase. This phase of the GCS development effort will produce the final working checkout system suitable for installation in an operational environment. The PHASE III system addresses such issues as redundancy, reactive and prerequisite sequencing, baseband/broadband technologies, timing and multiple communications busses.

Beyond PHASE III are actual development projects which will apply the architecture of the GCS to operational systems. These projects include the implementation of the Partial Payload Checkout unit in late 1989 and the Ground Data Management System in the near future.

3. DESIGN GOALS

The GCS project had very clear design goals from its inception. These goals were developed by applying lessons learned from implementation and operation of existing checkout systems. Fixing existing problems while providing an upgrade path for the future would be of utmost importance. Specific design goals for the GCS project include the following:

PROVIDE GENERIC CAPABILITY- The GCS would provide the building blocks for future ground checkout systems as well as other possible non-space related capabilities. The main GOAL of the GCS project was to provide the functional platform on which the other systems could be built. The GCS would act as the core checkout system and would be adapted to specific applications by adding layers of capability onto the GCS functions. This would allow an initial design and development effort to act as the bedrock for most future projects at KSC and elsewhere. Generic capabilities of the GCS would include the issuance of analog and discrete commands, response to exception conditions, user support environment, graphics workbench capabilities, archive and playback capabilities and other generic features.

PORTABILITY - Allow for the upgrading of the system in the future by adhering to industry standards throughout the GCS system. The portability issue applies to both hardware and software development efforts. Several aspects were considered in the software development effort, most importantly the use of UNIX[1] System V as the host operating system for all subsystems within the GCS. The use of the UNIX operating system provides several advantages, most importantly, UNIX is offered from multiple vendors and provides an interface definition which guarantees a large degree of portability of application software. This allows the system to avoid the problems of vendor dependance and application program obsolescence. Figure 1 shows the variety of hardware currently in use in the GCS system, illustrating the vendor independence of the system. This portability will be enhanced in the near future with the publication of the IEEE POSIX P1000 standard for portable operating systems. This standard will provide a vendor independent definition of portable operating system interfaces. All system and application programming for the GCS system would be done in the 'C' programming language. This was done to alleviate the current dependance on assembly language programming to achieve the performance necessary for real time checkout. In addition 'C' is well suited to the UNIX programming environment and much easier to maintain then the assembly language programs of the past.

1. UNIX is a registered trademark of AT&T

HARDWARE	ROLE
AT&T Single Board	DAM
AT&T 3B2	AP,DAP
AT&T 3B15	AP,DBS
AT&T 6300+	DP
SUN 3/110	DP
Apollo DN3000	DP
Apollo 5670	DP
Ironix	DAM
PT/VME 400	DAM
Plexus P75	ARS,DBS
Encore Multimax	DBS,AP
HP 825	AP
HP 350	DP
MASSCOMP	AP,DP
Gould PN6000	AP
IBM 9370	AP
INMOS Transputer	DAM
Mizar 68000	DAM

Figure 1. GCS Computer Hardware

Hardware obsolescence was also considered in the development of the GCS system. Current systems contain a large degree of obsolete or custom hardware which is difficult to maintain. In order to curb this problem in future systems the GCS would adhere to industry standard hardware interfaces. The use of standard Ethernet protocols for Local Area Networking, the SCSI interface for peripheral connection and standard VME bus architecture for custom architectures would be several of the standards used.

HIGH TECHNOLOGY - The GCS would also show a commitment to high technology developments which fell within the groundrules of portability. This high technology is apparent throughout the existing system and will become more apparent in the PHASE III system. The GCS utilizes state of the art Engineering Workstations, computer architectures and software development methodology to achieve its goals.

USER INTERFACE - The GCS would employ a modern user interface. Emphasis would be on the creation of a windowed environment based on mouse input. The user interface itself would attempt to adhere to the portability standards set forth in previous sections. Off the shelf software would be applied in as many cases as possible, graphics standards would be evaluated, XWINDOW windowing systems would be evaluated and high end graphics workstations would be used for display processing.

4. PHASE I SYSTEM

The Goals of the phase I system were to implement the rudimentary functions necessary to proceed with the more advanced checkout systems. The Specific goals of phase I were:

- Execute Ground Operations Aerospace Language interpretive code in a UNIX environment.
- Evaluate UNIX as a possible operating system for prototyping.
- Interface to existing control hardware to demonstrate compatibility.
- Evaluate GKS as a graphics standard for user inetrface displays.
- Utilize Ethernet as a means for inter processor communication.
- Demonstrate actual control using GOAL.

In order to implement the Phase I goals a minimum architecture was developed to support the development effort. Figure 2 shows the Phase I architecture. There were three subsystems developed for the Phase I system: The Application Processor(AP), the Display Processor(DP) and the Data Acquisition Module(DAM).

The Application Processor would be responsible for the execution and control of GOAL software. To understand the GOAL execution environment a brief overview of the GOAL language is required. GOAL is an extremely high level procedural language used for ground checkout operations. GOAL is edited and compiled on the LPS Central Data System to an intermediate language referred to as Interpretive Code. Configured Interpretive Code is then loaded into operational CCMS sets where it is executed by the GOAL Executor. The GOAL Executor is responsible for parsing GOAL Interpretive Code and interfacing with the CCMS operating system to provide services to GOAL. The GCS system implemented only the GOAL Executor portion of the GOAL environment. GOAL programs were still compiled on the CDS system and transferred, unmodified, to the GCS system. At that point a GCS GOAL Executor would handle the parsing and execution of GOAL interpretive code within the GCS environment. The GOAL Executor would be responsible for issuing commands across the Ethernet to DAM and DP subsystems to accomplish display update and end item commanding. The AP also provided keyboard command processing to allow real time interaction with the system. All programming of the GOAL executor and support software was done in 'C' to run under UNIX system V, Ver. 2. An AT&T 3B2 computer was used to support the AP role in phase I.

The Display Processor would be responsible for providing the user interface to the Application Processor. The DP would provide high resolution graphics displays which would be under the control of the GOAL programs executing on the AP and would allow interaction with the executing processor via keyboard and mouse interrupt processing. The Graphics Kernel System standard was adopted as the graphics standard for the DP system in phase I. This would allow the evaluation of GKS as a viable, portable graphics standard. The DP was developed on an IBM PC-XT to allow the evaluation of PC class units as Display Processors in an operational environment. In addition the PC's contained high resolution graphics and coprocessors to provide maximum performance in the DP area. The DP's, at this point, ran the MS- DOS operating system to allow for the widest possible vendor support for the DP development effort. The screen displays were developed in GKS to mimic the custom keyboard interface necessary to support GOAL application execution. All specialized keyboard functions were implemented with mouse point and click interfaces.

The Data Acquisition Module was responsible for the polling, linearization and exception monitoring of an attached data source. The DAM subsystem in Phase I was implemented to provide the same functionality as a Ground Support Equipment Front End Processor in the CCMS system. The DAM was basically and Analog/Discrete I/O processor containing 2 custom cards designed to interface with standard CCMS Hardware Interface Modules. The DAM would poll Analog/Discrete measurements on a cyclic basis, linearize the measurement if necessary, check for violations of pre-defined exception states and report these exceptions to the AP when they occured. The Phase I DAM was developed in an AT&T 6300 IBM PC compatible using the MS-DOS operating system.

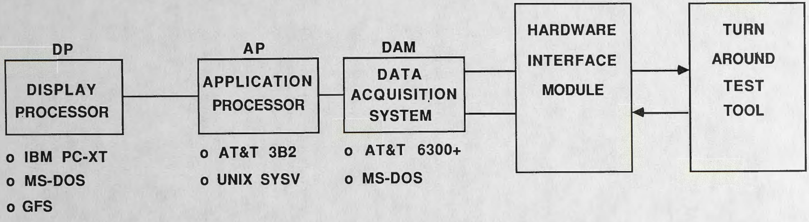


Figure 2

Phase I Architecture

The Phase I system was connected with standard Ethernet interfaces. The Ethernet transmission media and protocol would be evaluated to determine whether it was a valid means of interconnection within a real time checkout system. In order to minimize transmission overhead and maximize transmission speed the Ethernet Link Level transmission protocol would be used. The use of Link Level protocol enabled the implementation of custom protocols and the benchmarking of the Ethernet at the lowest possible interface level.

5. PHASE I RESULTS

The Phase I system was able to meet all the goals it set out to achieve. Approximately 70% of the GOAL operator set was implemented in the CGOAL executor developed for the UNIX system. GOAL program execution and control was identical to the CCMS operating environment, all operators involving the GSE interface control were implemented. UNIX proved to fit well into the development effort with no major problems. Performance of the GOAL executor was much less then desired using the UNIX system and an action was taken to improve this in the Phase II system. The implementation of an executor based system on a UNIX environment was an inefficient implementation at best. A proper implementation of the GOAL language would have to include the implementation of a streamlined executor or a compiled GOAL language. Interfacing to existing Hardware Interface Modules was also successful.

The Data Acquisition Module was able to poll and process analog and discrete measurements from the HIM interface. The DAM was also much slower than necessary to process actual data rates. The Phase I DAM could only process between 100 and 200 measurement per second as compared to existing polling rates of 10000 measurements per second in CCMS systems. Increase in processing power in the DAM area would be a goal in the Phase II system.

GKS was evaluated as the basis for graphics in the Phase I system and found to be unacceptable on two counts: 1 - GKS was extremely slow on the PC based workstations. A typical display would take over 1 minute to draw. 2 - GKS was not portable when based graphics software to larger workstations failed due to incompatibility in the 'C' language bindings.

Ethernet served very well in the Phase I system despite incompatibilities with the PC based Ethernet drivers. The data rates in Phase I, however, did not approach the bandwith of the Ethernet nor did they simulate actual data rates in existing checkout systems. Phase II would attempt to benchmark the Ethernet throughput capability of the architecture more closely than did Phase I.

6. PHASE II SYSTEM

The goals of the Phase II system were to build on the Phase I implementation and introduce several new subsystems and functions to the GCS configuration. The specific goals of Phase II were:

- Develop a subsystem to Archive and Retrieve command and measurement data during a live test.
- Develop a Data Base Subsystem to store and load measurement description data for the other GCS subsystems.
- Develop a windowing capability on the existing IBM PC/AT display processors.
- Develop an interactive means for display creation on the existing DP's as well as on high end engineering workstations.

- Create the capability to generate control and monitor programs in the 'C' programming language instead of GOAL.
- Develop a higher speed DAM using standard computer bus architectures.
- Evaluate industry standard SCSI bus standards as a means for computer to computer communication.
- Benchmark the Phase II system to determine command and measurement throughput.
- Demonstrate control of a responsive test article.

To develop these capabilities a greatly expanded testbed architecture was required. The AP and DP subsystems would still functionally exist in the GCS system. The Data Acquisition Module would be split into two subsystems, the Data Acquisition Processor(DAP) and the Data Acquisition Module(DAM). The DAP would be responsible for detecting and routing commands detected on the Ethernet to any one of 7 DAM's attached to the DAP via a Small Computer Systems Interface(SCSI) bus. This allowed the concentration of data packets to and from the DAM subsystems. DAM subsystems would be responsible for the polling of measurement data, linearization and exception limit checking. Any violation of exception limits would be sent to the DAP subsystem via SCSI and transmitted by the DAP to AP's in the system. The Data Acquisition Module would utilize VME bus technology and single board UNIX processors based on the 68000 family microprocessor. The Data Acquisition Processor would be based on the 3B2/400 from AT&T to maintain compatibility with the existing AP hardware. See Figure 3 for a Phase II system architecture overview.

The IBM PC based Display Processors would still exist in Phase II with the GKS based graphics software. In addition a rudimentary windowing environment would be developed on the PC DP's using Microsoft Windows, a commercial package. This windowing environment would allow the user to develop monitor only displays in an interactive fashion, that is, the user would not have to generate code to drive the display. In parallel, an effort to develop a usable graphics workbench on the Sun 3/110 workstation would be undertaken. This would allow the same interactive display creation techniques utilizing a commercial graphics workbench package. The Sun 3/110 would allow the user to create very high resolution displays and command from the same display. A commercial package called Data Views[2] was chosen to develop the workbench. This package already had much of the graphics workbench functionality necessary and integrated easily into several engineering workstations available for use. The Phase II DP workstation would have the capability of issuing commands directly to the DAP in the system to effect test article measurements. Windowing environments for the larger DP's would be examined in Phase II but not implemented. This was due to the fact that several windowing systems existed and provided little portability. Investigation into the XWINDOWS standard showed that workable versions of the XWINDOWS system would soon be available. XWINDOWS would provide a standard windowing interface across several workstation vendors and would provide the portability factor that Phase II was trying to maintain.

The Archival and Retrieval Subsystem would be introduced into the GCS architecture to provide long term storage and retrieval capability for live test data. The ARS subsystem monitors the Ethernet data bus for all command and measurement data traffic between subsystems and provides for filtered retrieval capability on recorded data. Retrievals range from very raw packet formats to processed individual command or measurement retrievals. This functionality is essential in checkout processing to enable playback and analysis of data should an unexpected situation occur. The ARS subsystem in Phase II was a Plexus P75 computer. This computer is a 68020 based machine running a

2. Data Views is a product of Visual Intelligence Corporation.

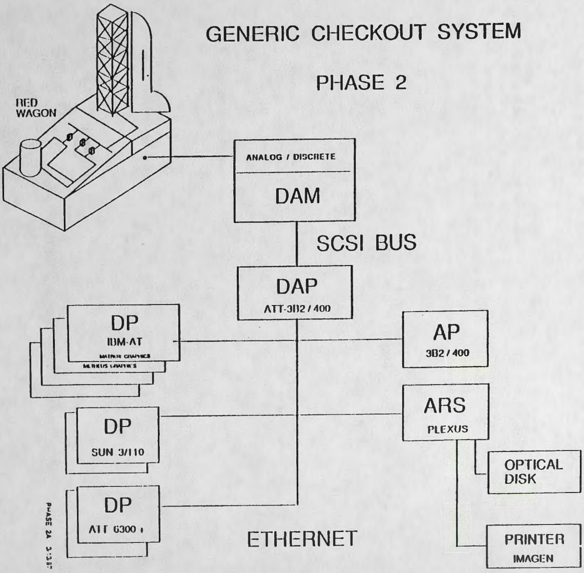


Figure 3
Phase II Architecture

UNIX operating system. To provide for very high capacity, long term archival storage optical disk storage devices were used. These devices provided for 1 Gigabyte of storage capacity per disk. Initially only 1 disk was used but this was later upgraded to 3. Archival storage to magnetic disk was also implemented to allow for higher speed, near real time retrieval which was not possible on the optical disk due to optical disk head speeds.

The ARS Subsystem would also be responsible for the creation and maintenance of relational database's in Phase II. This data base work included the reading, parsing and loading of relational data bases from user provided data description tapes. These tapes typically contain data concerning measurement description information for test articles which will be verified by the GCS system. The ARS was responsible for interpreting this measurement data and creating Measurement Description Tables which would be uploaded via Ethernet to the other subsystems in the GCS. This would eliminate the need to query the data base on line to determine measurement characteristics and routing information.

Application Programs in the AP would not be limited to GOAL applications. Higher level, industry standard applications would have to be adopted. The 'C' programming language was used to provide the end user with a standard programming language to use as a control and checkout language. Extension libraries were written to give 'C' equivalent functionality to the GOAL language. This enabled the GCS to act as a benchmarking testbed for compiled vs. interpreted language implementations.

The performance characteristics of standard UNIX were still a concern from Phase I. In order to attempt to increase performance of the AP and DAP subsystems a real time extension would be employed. The real time extension would overcome many of the scheduling and I/O problems in standard UNIX. This was done by providing non-degrading, absolute process priorities, high resolution timer services(to 10 milliseconds) and asynchronous Input/Output capabilities. The real time extension based system would also be benchmarked against the standard UNIX implementations to determine if a performance improvement was realized.

Along with the benchmarks for Real Time vs. Non-Real Time UNIX, benchmarks of the Ethernet throughput capabilities of the Phase II system would be conducted. These benchmarks would measure the maximum number of display and measurement commands which could be sustained in the system utilizing the Phase II hardware and software.

7. Phase II Results

The GCS Phase II system was completed in February of 1987 with great success. The Archival/Retrieval Subsystem was successfully implemented and rudimentary retrieval functions were developed. This gave the system the ability to retrieve archived test data from either a magnetic or optical disk for post test analysis. Concerns were raised concerning the ability of the ARS subsystem to maintain recording throughput to the optical disk while processing the network at peak rates. This problem would be investigated and benchmarked in Phase III. The Phase II ARS subsystem was successful in reading and recording all Ethernet packet traffic in Phase II demonstrations.

The Data Base processing portion of ARS functions was also successfully implemented in Phase II. The Data Base functions were capable of reading, interpreting and loading relational data bases for Payload Data Tapes and Payload Operation Control Center data tapes, work was begun on processing of the Shuttle Data Tape and will continue to be addressed in Phase III. The Data Base function was also capable of creating system and subsystem measurement data tables and uploading these tables to Data Acquisition Subsystems throughout the system.

The PC based windowing environment was created using Microsoft Windows. The windowing environment allows the user to create "monitor only" displays in an interactive fashion. The user can use mouse and menu driven interfaces to create displays based on the contents of the measurement description data base on the ARS system. The data is retrieved from the data base via Ethernet transactions. Several displays created in this manner could be displayed simultaneously using a tiled windowing format. This would, unfortunately, not enable every display window to show the entire contents of the display, some was hidden from view to allow the display of other windows. This was not an acceptable situation in an operational environment. In addition to this the slower GKS based systems were still in use on the PC display processors. These two problems prompted the decision to abandon the PC based display processors for more powerful engineering workstation based display processors.

Some of the Phase III graphics workbench functions were completed in phase II. These functions enabled the user to create high resolution graphics based displays in an interactive fashions. Displays would be capable of tracking and commanding measurements without the need for an associated control program. This enabled the user to create displays and execute them without having to develop software. The screen update rates on this type of workstation were visually much faster then the previous GKS/PC combination. The use of the Data Views package to create this graphics workbench also provided several benefits. The package was available on all of the major workstations and displays created on one workstation were 100% portable to other workstations regardless of the workstation manufacturer. Test of portability were performed with Sun 3/110, Apollo DN300, HP9000/350 and Masscomp workstations with good success.

Libraries were created for the 'C' programming language to enable the user to use 'C' as a command and control language. These libraries allowed the user to issue commands to affect measurements, respond to out of limits conditions, control display attributes and interface to the user at a remote DP workstation.

Higher speed Data Acquisition Modules were developed based on the 68000 family microprocessor and VME bus architecture. These Data Acquisition modules polled analog and discrete measurements using standard AD/DA cards available commercially. The Data Acquisition Module would also check all limit violation conditions, linearize the data, check for significant change conditions and report findings to the DAP via SCSI interface. The DAM in this phase was processing measurements at between 1000 and 2000 measurements per second. Still well short of the 10000 Measurement per second target. The development of very high speed data acquisition modules would be the goal in the Phase III system. The Data Acquisition Modules in this Phase were also still based entirely on the UNIX operating system on a single 68010 processor.

The SCSI bus connecting the DAP and DAM subsystem was evaluated in this phase and several problems were discovered. The SCSI is basically a peripheral connection bus and does not adapt well to computer to computer communications tasks. Driver design and implementation were necessary to cause the SCSI to work in this mode. Also, there are some distance limitations with the SCSI bus, about 24 Meters maximum with a differential bus. It would be necessary in some operational environments to place the DAP and DAM subsystems several hundred meters apart. Because of these reasons it was decided to investigate different DAP to DAM bus architectures in Phase III.

Benchmarks were conducted on the Phase II system. These benchmarks served two major functions 1) measure the maximum throughput of the phase II system based on number of commands processed per second and 2) compare system performance using Real Time Extensions and standard UNIX. Benchmarks of all commands in the 'C' libraries were run with 1, 2 and 3 concurrent tests running. The use of Real Time Extensions to UNIX showed a marked performance improvement over standard timeshare UNIX. Taking as a representative example the command to set a Discrete value, Figure 4 shows the actual commanding times for 1, 2 and 3 concurrences for the UNIX and UNIX/RTE system. Note that the times represent about a 45% - 60% performance improvement when RTE functions were employed. Later improvements in the AP and DAP software effectively halved the commanding time needed for the issuing of discrete commands. This brought the commanding time to an average of about 1/10th of a second.

CONFIGURATION	PROGRAM 1	PROGRAM 2	PROGRAM 3
TIMESHARE 1	.432 SECS	-	-
RTE 1	.238 SECS	-	-
TIMESHARE 2	.836 SECS	.781 SECS	-
RTE 2	.24 SECS	.24 SECS	-
TIMESHARE 3	.569 SECS	.542 SECS	.525 SECS
RTE 3	.293 SECS	.272 SECS	.310 SECS

Figure 4. RTE VS. UNIX, Set Discrete Value

8. Phase III

The Phase III GCS implementation will be the most ambitious to date. The culmination of the Phase III development effort will be the core set of checkout functions necessary to support an actual checkout system development effort. All GCS subsystems will be implemented and fully functional. The Phase III system is currently in the design Phase with an expected completion date of July 1, 1988. The goals of the Phase III effort are to build the Phase II system into an operational checkout system. The specific goals for Phase III are:

- Implement a flexible data passing philosophy which enables easy integration of new hardware modules.
- Support payload unique Data Acquisition Modules(Payload Data Interleaver, Payload Signal Processor, GSE).
- Support multiple Ethernet interfaces per Subsystem to allow separation of critical and non-critical data traffic.
- Utilize the TCP/IP protocol for non-time critical data transfers.
- Implement higher performance subsystems to allow for Phase II benchmark comparisons and identify minimum machine classes for future projects.
- Implement a working Data Base Subsystem for Configuration Management, system loading and configuration and Data Base Management.
- Develop control logic sequencing, both reactive and prerequisite.
- Demonstrate subsystem redundancy and active-standby switchover capability at the DAP level.
- Complete the User Support Environment on the DP workstations as well as the graphics workbench.

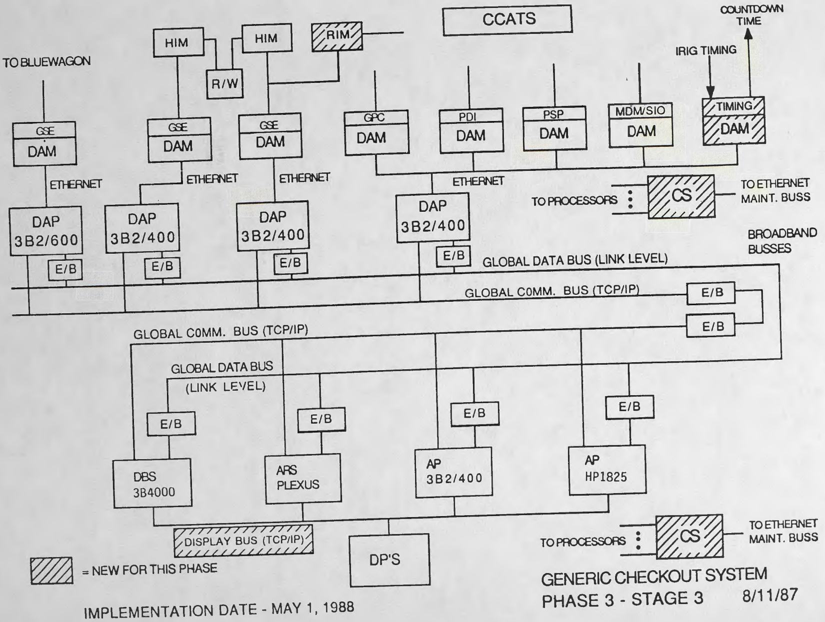

CCATS
COUNTDOWN TIME
IRIG TIMING
TO BLUEWAGON
HIM
HIM
R/W
RIM
GSE DAM
GSE DAM
GSE DAM
GPC DAM
PDI DAM
PSP DAM
MDM/SIO DAM
TIMING DAM
ETHERNET
ETHERNET
ETHERNET
ETHERNET
DAP 3B2/600
DAP 3B2/400
DAP 3B2/400
DAP 3B2/400
E/B
TO PROCESSORS
CS
TO ETHERNET MAINT. BUSS
BROADBAND BUSSES
GLOBAL DATA BUS (LINK LEVEL)
GLOBAL COMM. BUS (TCP/IP)
GLOBAL COMM. BUS (TCP/IP)
GLOBAL DATA BUS (LINK LEVEL)
DBS 3B4000
ARS PLEXUS
AP 3B2/400
AP HP1825
DISPLAY BUS (TCP/IP)
DP'S
= NEW FOR THIS PHASE
IMPLEMENTATION DATE - MAY 1, 1988
GENERIC CHECKOUT SYSTEM
PHASE 3 - STAGE 3 8/11/87

Figure 5

Figure 5 shows the proposed Phase III hardware layout. The most prevalent feature of the new Phase III architecture is the use of the multiple network configuration. There will be three separate networks in the Phase III configuration, these are the Display Bus(DB), the Global Data Bus(GDB) and the Global Communications Bus(GCB). The Local Display Bus is used to allow the DP workstations to receive necessary measurement and display information without increasing traffic on the data and command buses. The display processor in Phase III will be totally autonomous, that is, the control programs running in the Application Processor will not be responsible for updating DP displays, this is done entirely by the DP workstation graphics workbench. The Display Bus is also used for remote login and file transfer to and from the AP, ARS and DBS subsystems. this allows the user to do software creation in a distributed environment providing more flexibility. The Global Data Bus is an Ethernet bus but is treated in a uni-directional fashion. Exception and Significant change data transmitted from the DAP subsystems travels over this bus. No command or data transfer from the AP subsystems is placed on the GDB. The GDB is the most critical bus system, the data travelling on this bus must be least prone to collision and retry then the other buses. The Global Communications Bus carries traffic having to do with commanding and subsystem status. Commands from automated and keyboard command sequences travel over this bus to the destination DAP subsystem. Command acknowledgments are also transmitted over this Ethernet connection. The Global Data Bus and Global Communications Bus both will utilize Ethernet to Broadband converter boxes to allow an adaptable system architecture. The GCS Phase III system will be capable of supporting several simultaneous tests by reconfiguring the broadband connections to place different tests on different broadband channels. The Broadband system will support up to 14 simultaneous channels.

The Data Base Subsystem will be developed using a commercially available relational data base. To maintain the portability of the system the SQL and ESQL structured data base query languages will be used to develop and maintain data base files. The ESQL(Embedded Structured Query Language) will be used in conjunction with the 'C' programming language. The DBS will be responsible for reading, parsing and storing data pertaining to measurements and commands as they are supplied from various data tape sources. These sources can define measurements for single payloads to entire missions. The measurement data will be kept in a measurement descriptor data base. From this data base system Measurement Descriptor Tables will be built for the other subsystems to use operationally. These tables are uploaded via the Global Communication Bus to all subsystems supporting the current test. Using prebuilt tables instead of querying the DBS for data will serve to increase system throughput and decrease measurement processing time. The DBS will also be responsible for processing data retrieval requests from remote DP workstations. Data retrieval requests are used to filter and display data previously recorded by the ARS subsystem. The DBS is responsible for retrieving data from local optical disks and sending it in processed format to data display programs running at the DP. Data retrieval displays will allow for filtering of commands or measurements, displaying through graphics and presenting trending data on one or more measurements simultaneously. The DBS will be implemented on an AT&T 3B4000 multiprocessor. A goal in phase III will be to determine the type and size of machine necessary to develop the DBS subsystem. The DBS will also be the subsystem for all system and subsystem configuration management(CM). CM duties of the DBS will include maintaining user and system source libraries, restricting user access to CM source libraries, maintaining all executable files for all GCS subsystems and loading executable files to subsystems during system configuration.

The TCP/IP protocol will be adopted for many non-time critical data transfers. This would include system configuration and loading, user application file transfers, remote login capabilities for software development and some subsystem commanding. The TCP/IP protocol was selected due to its wide acceptance on UNIX based processors and the rigorous standard in place for TCP/IP transactions. TCP/IP will not be used for commanding and measurement data passing due to the routing overhead within the TCP/IP packet structure. Ethernet Link Level access will continue to be used as it allows the creation of tailored data interchange packets which can minimize overhead.

To insure the flexibility and upgradeability of the GCS architecture a new data passing philosophy will be implemented in the Phase III system. Data Acquisition Modules in Phase III will be responsible for all processing of raw data prior to delivery to the rest of the GCS system. This enables standardization of all communications to and from the DAM interfaces. The standardization of this interface will limit the new software development efforts to the DAM area when new interfaces are developed and integrated into the system. Current systems do not processes the data prior to delivery to the application program. Measurement data for analog and discrete measurements are kept in raw count format and processed by the application program when required. Pre-processing data decreases the complexity of application program data libraries necessary to perform checkout functions. Pre-processed data are stored in shared common data areas in the AP subsystem in IEEE floating point format for analogs and a compressed format for discrete measurements.

A large part of the work in Phase III will be the implementation of the unique Data Acquisition Module interfaces necessary to process actual payload commands and telemetry. There are 4 new interfaces planned for the Phase III system. The first, The Payload Data Interleaver(PDI), DAM is responsible for processing 128 kilobit PCM data streams containing Analog and Discrete data points. In processing this data the PDI DAM will be responsible for acquiring the data, decomutating the data into individual measurements, checking the current data values against 3 separate sets of exception limits, linearizing analog measurements to 7th order polynomial expansions and converting the data to the standard format the GCS system is expecting. The volume of data would overwhelm the single UNIX processor in the DAM in Phase II, to improve on this several new technologies will be used in the Phase III Data Acquisition Modules. First and most importantly, is the parallel processing approach. The PDI DAM will use 4 separate processors to accomplish its task; the UNIX engine, the Filter card(s) and the VME bus link card. The filter card will be responsible for most of the work within the DAM. The filter card will be responsible for acquiring, linearizing, exception checking and converting the data. To accomplish this the filter card is being designed with Transputer chips. A Transputer chip is a very high speed(approximately 12 MIPS per chip) processor which is built for parallel processing applications. The filter card will utilize 2 INMOS Transputers connected via high speed serial link adapters to accomplish the filter functions. The filter card will rely on the VME bus link card to communicate with the VME bus which is the host for the DAM architecture. The Bus link card will be based on the 68010 microprocessor and will be purchased off the shelf. The Bus Link Card will be responsible for transferring exception and significant change notifications from the Filter card to the Ethernet interface card which is used to communicate with the DAP subsystem. The UNIX engine within the system will be responsible for initial DAM subsystem loading and transferring some low priority commands to the Bus Link Card. Similar architectures will be employed in the remaining DAM interfaces; Payload Signal Processor(PCM 128Kb Output), GPC(Orbiter General Purpose Computer Simulator) and GSE(Ground Support Equipment control and monitor). In the case of the GSE DAM the DAM will be responsible for polling analog and discrete measurements instead of reading a serial stream of data. The Phase III GSE DAM will process polling of measurement at the rate of 10000 measurements per second or better as a target. This will duplicate existing polling rates in the CCMS subsystem of the Shuttle Launch Processing System.

Implementation of higher performance subsystems will be a priority in Phase III. The inclusion of higher performance, varying architecture machines in this phase will allow the comparison of Phase II benchmarks with phase III benchmarks to determine where the system bottlenecks and critical performance areas reside. It will also show the performance of varying computer architectures in several subsystem roles. Parallel processing, RISC architectures and standard uni-processor systems will be developed and benchmarked in many GCS Subsystem roles. Systems from multiple vendors will be used to demonstrate subsystem software portability and upgradeability.

The User Support Environment will be completed in Phase III. The User Support Environment(USE) will support a graphics workbench, mouse driven user interface, multiple windowing system and software development tools. The graphics workbench will provide the user with the tools necessary to create high resolution, active monitor displays. These displays will be capable of monitor and command functions when displayed without the need for the user to generate code to run them. This allows creation of monitor only displays at any time during operations should it become necessary to monitor new data sets. The graphics workbench will be tied directly to the DBS subsystem for retrieval of measurement descriptions and data formats in order to create displays interactively. The graphics workbench will be icon driven and provide a library of standard data symbols for the user to employ, capabilities will also exist for custom graphics icon creation, display zoom features, activation of sub- displays and pull down menu creation.

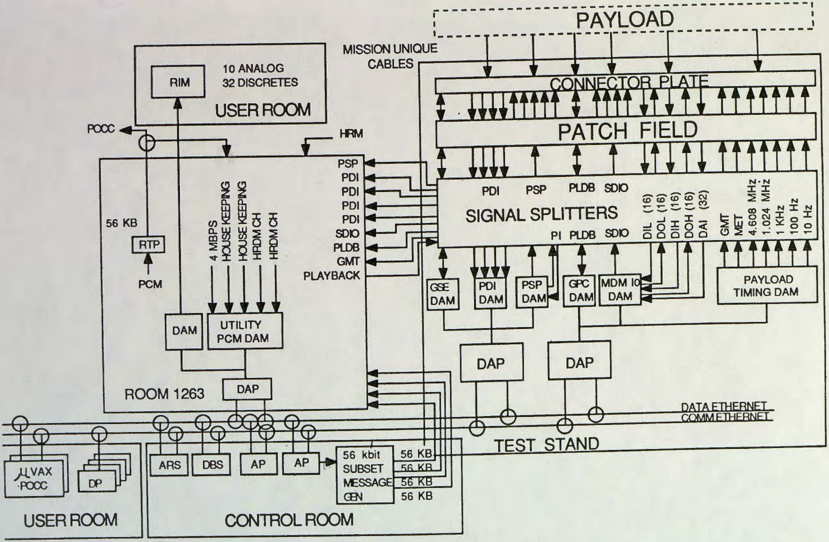
The graphics workbench will be one of several tools available to the user via the user support environment. The USE will provide the user with an interactive means of navigating through the USE tool set. This interface will employ window driven interfaces to allow for the use of multiple displays simultaneously. The USE will also provide editors, network transfer tools and the operational DP environment. The Operational DP environment is used during actual control and checkout operations for the display and control of live test data. Multiple graphics displays may be activated and commands may be sent through active displays or a keyboard command processor. The OPS environment will also contain system message windows, personal message windows, exception monitoring displays and remote system login capability.

Standards will again be the norm in the User Support Environment. Several windowing standards, XWINDOWS, NEWS, DIALOG, will be evaluated as the standard to be used throughout the DP development environment. All DP workstations will run the UNIX operating system as does every other subsystem within the GCS.

9. FUTURE OF THE GCS CONCEPT

After the completion of Phase III the GCS architecture will take its first steps into the operational world. The initial application of GCS architecture will be in the Partial Payload Checkout Unit(PPCU) project currently in work at KSC. The PPCU will employ the same architecture as the GCS Phase III system but will add those functions necessary to provide unique PPCU functions. This supports the GCS function of being a core set of functions to be built on to support specific programs. The architecture of the PPCU system is shown in Figure 6. The basic architecture of GCS is there along with the PPCU unique aspects such as the 56Kbit subset message generator, the Remote Interface Module and the Payload Timing DAM. These functions are unique to the PPCU system but will interface easily to the GCS system as standards will be maintained in these systems also. The PPCU system will enter the preliminary design phase in February of 1988 and is scheduled to be operational in early 1989.

The other major projects which will be based on the GCS architecture will be the Ground Data Management System(GDMS) and the replacement for the Control Checkout and Monitor(CCMS) portion of the Shuttle Launch Processing System. The GDMS system will be used to process Space Station elements as they are delivered to KSC for checkout prior to launch. The GDMS system will prove to be much larger then the PPCU implementation and will fully develop the GCS concepts of multiple test support and reconfigurability. The CCMS system will also be a very large system and will have the task of integrating with existing launch processing hardware and software to enhance current capabilities.



PPCU SYSTEM

Figure 6

10. CONCLUSIONS

The GCS has proved to be a valuable testbed in evaluating software and hardware architectures for ground checkout operations. The GCS has provided a solid foundation for the technologies necessary to develop successful systems. The use of UNIX has proven to be very successful and should provide a large degree of portability in system and application software for future upgradeability. The high technology advancements in use in the GCS prove that non-custom hardware and software can provide a basis for specialized checkout systems and reduce the design and development efforts of these systems. The User Support Environment of the GCS should provide for a measurement of productivity improvement when using advanced user interface techniques for system and application program development. The GCS as a whole shows the interconnectability of various vendors hardware and software into a system which is portable, upgradeable, expandable, reconfigurable and which should fully support the needs of the space program at KSC for the foreseeable future.