RECONFIGURABLE GATEWAY SYSTEMS FOR SPACE DATA NETWORKING

Don Davis, Toby Bennett, Jay Costenbader

TSI TelSys, Inc.
7100 Columbia Gateway Drive
Columbia, Maryland 21046

ABSTRACT

Over a dozen commercial remote sensing programs are currently under development representing billions of dollars of potential investment. While technological advances have dramatically decreased the cost of building and launching these satellites, the cost and complexity of accessing their data for commercial use are still prohibitively high.
This paper describes Reconfigurable Gateway Systems which provide, to a broad spectrum of existing and new data users, affordable telemetry data acquisition, processing and distribution for real-time remotely sensed data at rates up to 300 Mbps. These Gateway Systems are based upon reconfigurable computing, multiprocessing, and process automation technologies to meet a broad range of satellite communications and data processing applications. Their flexible architecture easily accommodates future enhancements for decompression, decryption, digital signal processing and image / SAR data processing.

KEY WORDS

Telemetry Protocol Processing, Satellite Communications, Reconfigurable Computing, Real-time Processing, Test & Simulation

INTRODUCTION

A host of remote sensing programs will be deployed in the next five years to meet the demand for high resolution imaging data for commercial and scientific applications. These applications include environmental monitoring, precision farming, urban planning, resource exploration management, and tactical reconnaissance. While technological advances have decreased the costs of building and launching satellites, the costs and complexity of timely access to satellite data are still prohibitively high.

Remotely sensed image data with the highest commercial value will have a one to three meter spatial resolution. The resulting telemetry data stream characteristics (i.e., 100+ Mbps) rival those of NASA's most ambitious program, the Earth Observing System. Historically, NASA's cost to operate and implement high data rate telemetry systems has been tens to hundreds of millions of dollars.

Key requirements driving future low-cost ground stations include:

- Multi-mission support
- Real-time or near real-time processing and distribution
- Autonomous remote operation
- Connectivity to commercial RAID and tape storage devices
- Interoperability with commercial network environments
- Flexibility to meet changing / future requirements

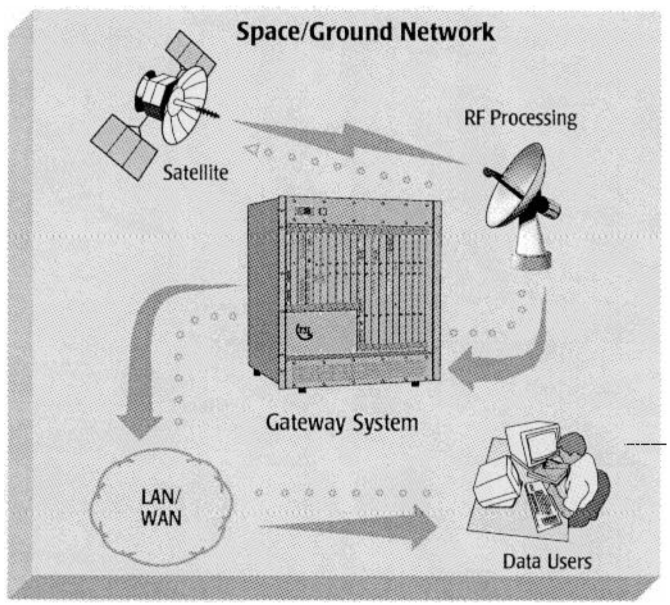

Figure 1 - Gateway System Connectivity

The challenges of many future ground station requirements are handled in TelSys Gateway Systems which perform the sophisticated telemetry and networking functions required to interconnect local and wide area networks to satellite communications networks. Figure 1 depicts this interconnectivity.

TelSys has developed an approach which meets these requirements by using a novel mix of reconfigurable computing technologies, object oriented embedded real-time software, and object oriented system control and management software.

Reconfigurable computing involves the use of in-circuit reprogrammable hardware elements to provide the real-time processing of data. By using an array of these dynamically reconfigurable hardware elements with object oriented real-time and workstation software, virtually any processing requirement can be accommodated. This paper describes how these technologies are used to implement systems of unparalleled performance and functionality at a very low cost.

GATEWAY SYSTEMS

Gateway Systems are an integration of three core elements: a Reconfigurable Computing Platform (RCP), Local Control Software (LCS) and Gateway Management Software (GMS). These three elements are all highly flexible and configurable to handle a wide variety of processing needs. The RCP is a standalone hardware platform which provides interconnection of the various hardware elements and physical interfaces for the processing of high speed data. LCS provides the real-time status and control software. In addition it manages the configuration of the RCP by providing the mapping of hardware processing algorithms to available hardware resources. GMS provides remote autonomous control of one or more RCP platforms.

Reconfigurable Computing Platform

A block diagram of the RCP is shown in Figure 2. The platform is based on the industry standard VMEbus. This bus, using VME64 extensions, provides a theoretical maximum of 80 Mbytes/second transfer rate. However, arbitration overhead on this bus can substantially reduce the actual useable bandwidth. Therefore, this bus is used primarily for the transfer of control and status information between cards.

The real-time transfer of high speed data between cards is handled using either the standard RACEway Interlink and/or the High Speed Backplane. For the 6U platform, only RACEway is used. This ANSI/VITA standard provides up to 160 Mbytes/second point to point connections between cards. The High Speed Backplane is used for 9U form factor cards and can work in conjunction with the RACEway Interlink. The High Speed Backplane uses a third backplane connector providing multiple high speed parallel paths between data processing cards. Six master channels, each capable of sustained data rates of over 320 Mbps, can be further subdivided into independent subchannels providing over 5 Gbits/second of aggregate transfers.

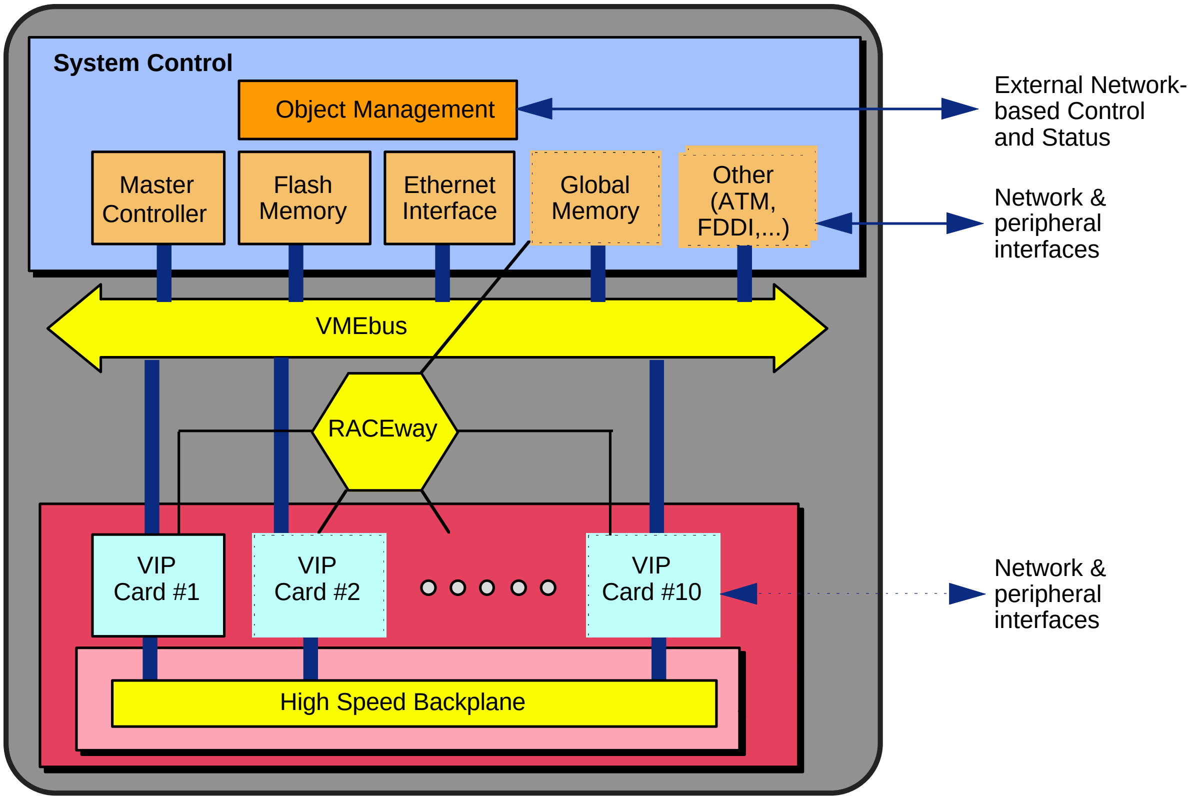

Figure 2 - RCP Block Diagram

The System Control portion of the Gateway System consists of a number of baseline functional elements: the Master Controller, Flash Memory and an Ethernet Interface. The Master Controller card acts as the VMEbus arbiter and oversees all activity in the unit. An Ethernet interface provides network access and control via remote workstation-based Gateway Management Software. The system can also be controlled locally via a terminal interface. Flash memory, residing on front-panel removable PCMCIA modules, stores all the system boot-up firmware and application code. This allows for very easy field maintenance and upgrade. Other cards may be added as needed including global memory for data buffering and additional network and peripheral interface cards. Network and peripheral interface cards support standards such as ATM, FDDI, SCSI-2, HiPPI and Firewire.

Real-time data processing is performed using Virtual Information Processor (VIP) cards. Up to ten of these cards can be accommodated in a RCP depending on the system processing requirements. The cards feature a generic processing engine based on reconfigurable computing technology. The processing logic for all high speed data manipulation is performed in an array of Xilinx 5200 Series FPGA parts. The combination of these parts provide up to 108,000 reconfigurable gates for algorithm implementation. In addition, there are a variety of memory elements to provide data buffering, temporary data storage and look up tables including six 8KByte First In First Out (FIFO) memories, two 16KByte Dual-Ported Random Access Memories (DPR) and two Single In-Line Memory Modules (SIMM) which can accommodate up to 8MBytes of Static Random Access

Memory (SRAM). The processing algorithms are dynamically downloaded into the RCP at system boot-up, or during run-time from the PCMCIA flash memory or the network. Physical input and output interfaces for these cards are provided by plug-in modules based on the industry standard Peripheral Component Interconnect (PCI) Mezzanine Card (PMC) format. The 9U VIP subsystem supports three PMC modules while the 6U VIP supports up to two PMCs. PMC modules are used to implement CPUs, DSPs, serial interfaces, ATM, SCSI-2, FDDI, NTSC, as well as complex functions such as error corrections or data compression.

A given processing subsystem, known as a VIP "instance," is defined by particular selection of downloadable hardware bitstream configuration, software modules and PMC modules (see Figure 3). One example of an instance is the Front-End Processor subsystem, which performs frame synchronization, CCSDS Reed-Solomon error correction and virtual channel sorting. Other instances include the Dynamic Simulator, Forward Link, Common Data Link (CDL) Interface, Video Processor, Data Verifier, NASCOM Interface, and Line Processor. The use of downloadable hardware functionality and multiple plug-and-play PMC modules allows the implementation of virtually any satellite communications processing function. There are numerous advantages to this approach, including hardware processing speeds with the flexibility of software solutions, simplified field hardware upgrades, and fabrication economies of scale.

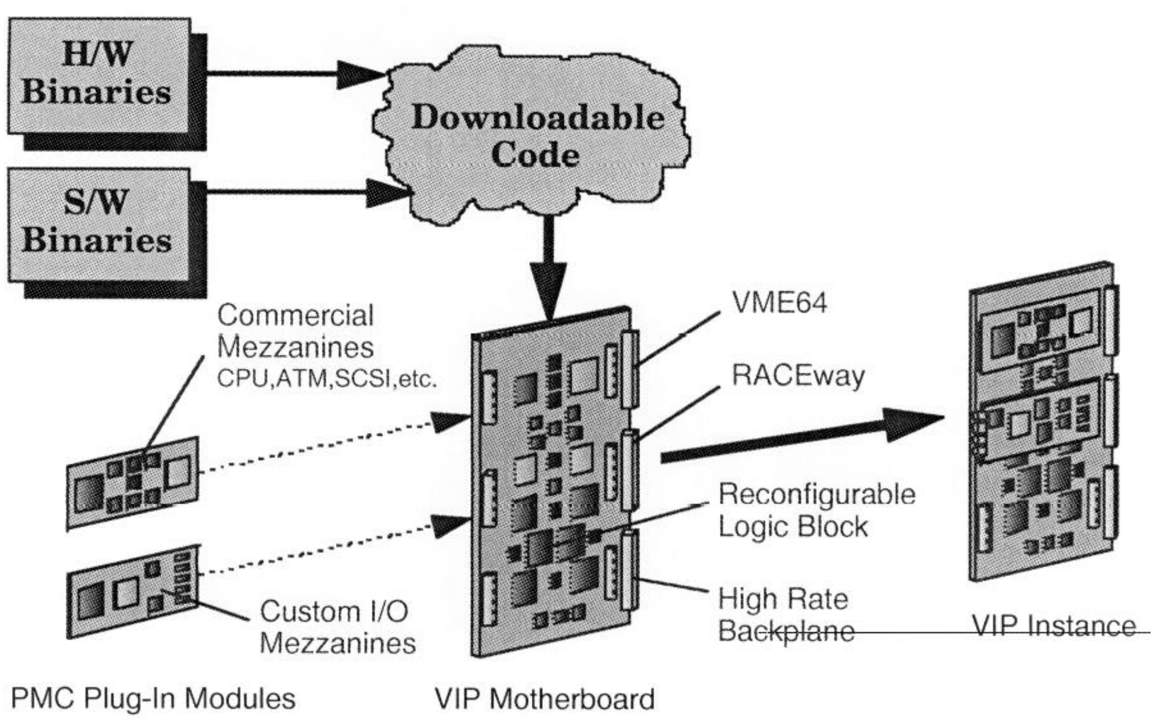

Figure 3 - VIP Instance Components

The RCP is currently hosted on both the VME 6U and 9U platforms. A desktop verison based on the PCI bus is under development. This provides a choice of platforms at various function, price, and performance points while maintaining hardware and software binary compatibility of VIP instances. Figure 4 shows the evolution of the Gateway Systems.

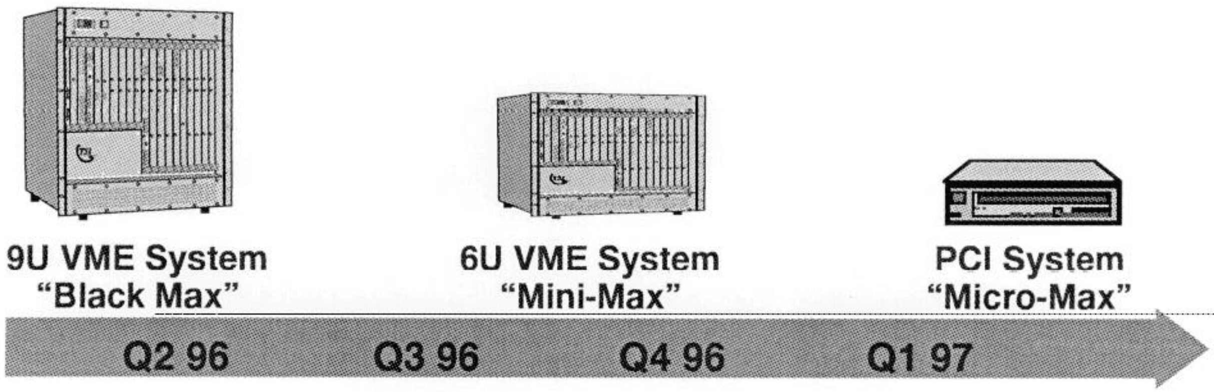

Figure 4 - Gateway System Evolution

In addition to platform evolution, the library of instances is constantly growing and the library of PMC modules, which are usable across all platforms, are being enhanced to include new functions. PMC modules for Rice decompression, MPEG-2 decompression, and TAXI are currently planned. Integration of several third-party modules are also planned for Digital Signal Processors (DSP), Fibre Channel and PCMCIA interfaces.

Local Control Software

In addition to the VIP hardware elements, a key feature of a Gateway System is that it provides a high-performance software environment that supports dynamic reconfiguration. The Local Control Software (LCS), which is layered on top of Wind River Systems' VxWorks real-time operating system, provides a client-server architecture with an open framework for the application of generic Gateway Systems and custom, application-specific subsystems.

LCS, shown in Figure 5, includes two categories of reusable software components: run-time system service components and general-purpose subsystem software component templates.

In the way of run-time system services, LCS includes a number of software components that provide services for:

- Controlling the allocation of subsystems across VIP hardware resources
- Providing a general-purpose, inter-subsystem communication mechanism

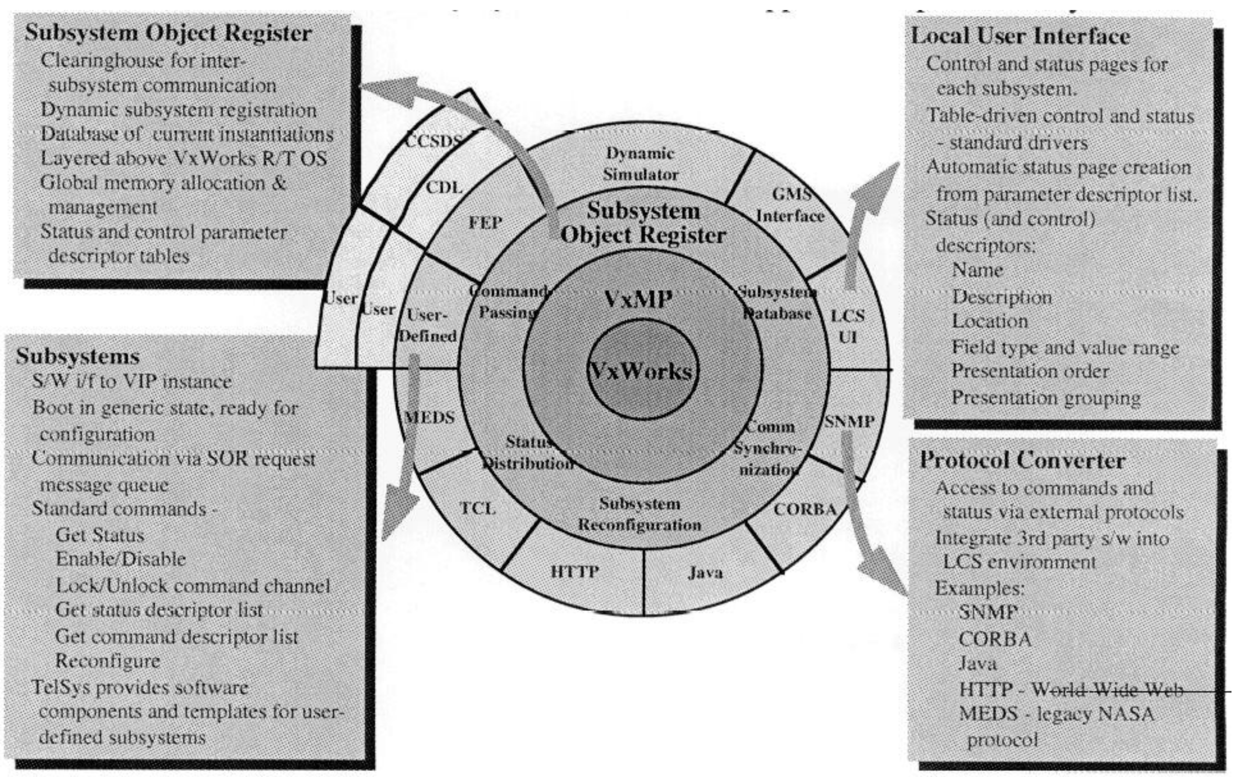

Figure 5 - The Layered Architecture of LCS

- Providing a subsystem locator database
- Monitoring operation of subsystems
- Managing the allocation of memory and other limited resources across subsystems

These elements of the LCS software environment provide a two-tier client-server environment for management and operation of real-time satellite data networking applications.

In addition, LCS also provides a number of software modules that can be used verbatim to control and monitor many application software subsystems and, when necessary, can be extended for specific applications. Thus, an application developer can build a software subsystem simply by implementing application-specific subroutines and linking them with the OTS components provided in the LCS libraries. These application-specific subroutines effectively overload generic subroutines that are already implemented in the LCS libraries. Some of the LCS subroutines include:

- subsystemInit() -- Initializes subsystem hardware and software resources.
- subsystemKillResources() -- Cleans up and releases resources.

- subsystemProcessCommand() -- Handles subsystem-specific commands that are not handled automatically by the LCS command handler.

- subsystemGetStatus() -- Request status from the subsystem.

Since the processing elements of a Gateway System are based on the same configurable VIP hardware elements, the LCS software provides a means of boot-strapping the system to a neutral state, so it can then be configured for application-specific space data networking operations. What this means is that the actual configuration of a system is determined at run-time, rather than a-priori, from configuration files that specify which processing instances are loaded into each VIP hardware element and which software subsystems are allocated for execution.

Upon system boot, a generic software subsystem is assigned to each VIP hardware element. A series of master controlling elements then assigns application-specific subsystems (and VIP processing instances) to hardware elements, based on run-time interpreted files. This set of master controlling elements includes the Hardware Resource Database, Master Instantiator, Subsystem Manager, Status Buffer Manager, Message Queue Manager, Alias Manager and Local Instantiator.

The Hardware Resource Database maintains a database of hardware resources that is referenced when allocating subsystems to hardware elements, since a specific subsystem may require a VIP with a specific type of hardware resource (e.g., an I/O mezzanine). The Master Instantiator has primary responsibility for allocation, creation, and deletion of subsystems for both VIP instances and software-only subsystems. The Subsystem Manager maintains a database of information about all active subsystems that can be queried by any subsystem. The Status Buffer Manager is responsible for allocation of semaphore controlled memory areas to subsystems. This memory is typically used to return processing results or status. The Message Queue Manager is responsible for allocation of inter-subsystem communication message queues. The Alias Manager maintains a database of alias names for various hardware and software resources, subsystems, commands, and memory areas, so subsystem names need not be hard-coded. The Local Instantiator incorporates the generic software that initializes a subsystem and manages interaction between the application-specific software and LCS.

The LCS software framework provides an extensible, client-server environment for building and operating embedded VME systems that utilize reconfigurable computing technologies. This framework is combined with VIP reconfigurable hardware elements to provide a set of off-the-shelf tools and resources that allow end-users to develop their own hardware-accelerated custom applications, along with the software necessary to control and communicate with these applications.

Gateway Management Software

A natural extension to the Gateway Systems platform is the Gateway Management Software (GMS) (see Figure 6). GMS provides a complete package for the operation of space data networking applications including autonomous network-based control of one or more Gateway Systems, interacting with an LCS subsystem for external control and monitoring interfaces. GMS acts as the system's network manager, maintaining spacecraft activity schedules, performing fault detection, isolation, and recovery (FDIR), and reporting status to higher level control elements that may be at remote sites. Functions of the GMS include:

- Autonomous, scheduled control of Gateway Systems and supporting elements
- Automated telemetry data distribution
- Data quality statistics accumulation and logging
- Local or remote GUI
- Integrated expert system-based control and automation
- Extensible, client-server architecture
- Application Programming Interface (C and C++)
- SNMP-compliant network management interface (i.e., agent)

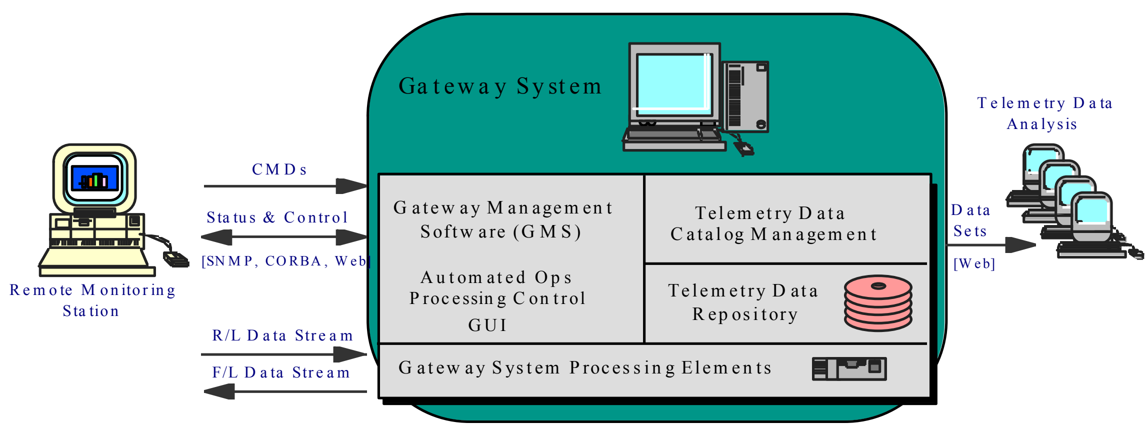

Figure 6 - Gateway System Features and Interfaces

The GMS software system provides a client-server architecture that allows for assembly of a space data networking operations system from a set of off-the-shelf software components that surround a central inter-process communications mechanism. The client-server nature of GMS provides an open architecture that allows for easy integration of application-specific capabilities. An end-user may extend the capabilities of GMS to his/her application using an Application Programming Interface (API). This API instantiates a series of software idioms that allow the end-user to integrate his/her application into GMS, since all GMS components use this same API. Additionally, GMS provides a common means of communicating with user-developed applications from a remote control workstation via a SNMP agent.

CONCLUSION

A worldwide explosion of advanced remote sensing and space science programs will drive the demand for more advanced ground station processing systems supporting a much broader, and more mobile customer base. These systems must provide greater performance and functionality while being significantly lower in cost and size. Gateway Systems meet the next generation program needs. They incorporate the best of current object-oriented hardware and software technologies including on-the-fly reconfiguability in a fraction of a second.

ACRONYMS

ANSI	Amrican National Standards Institute
ASICs	Application Specific Integrated Circuits
ATM	Asynchronous Transfer Mode
CCSDS	Consultative Committee for Space Data Systems
CDL	Common Data Link
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
DPR	Dual Ported Ram
DSP	Digital Signal Processing
FDDI	Fiber Distributed Data Interface
FDIR	Fault Detection, Isolation and Recovery
FIFO	First-In First-Out
GMS	Gateway Management Software
GUI	Graphical User Interface
HiPPI	High Performance Parallel Interface
LAN	Local Area Network
LCS	Local Control Software
Mbps	Megabits per second
NASA	National Aeronautics and Space Administration

NTSC	National Television Systems Committee
PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
PMC	PCI Mezzanine Card
RCP	Reconfigurable Computer Platform
SCSI	Small Computer Systems Interface
SIMM	Single In-Line Memory Module
SNMP	Simple Network Management Protocol
SRAM	Static Random Access Memory
TAXI	Transparent Asynchronous Transmitter/Receiver Interface
TDM	Time Division Multiplexing
VITA	VMEbus International Trade Association
VIP	Virtual Information Processor
VLSI	Very Large Scale Integration
VME	Versa Module Eurocard
WAN	Wide Area Network