Remote Diagnosis of the International Space Station utilizing Telemetry Data

Somnath Deb[a], Charles Domagala[a], Sudipto Ghoshal[a], Ann Patterson-Hine[b], and Richard Alena[b]
[a]Qualtech Systems, Inc., Suite 501, 100 Great Meadow Road, Wethersfield, CT 06109
[b]NASA-ARC Research Center, Moffett Field, CA

ABSTRACT

Modern systems such as nuclear power plants, the Space Shuttle or the International Space Station are examples of mission critical systems that need to be monitored around the clock. Such systems typically consist of embedded sensors in networked subsystems that can transmit data to central (or remote) monitoring stations. At Qualtech Systems, we are developing a Remote Diagnosis Server (RDS) to implement a remote health monitoring systems based on telemetry data from such systems. RDS can also be used to provide online monitoring of sensor-rich, network capable, legacy systems such as jet engines, building heating-ventilation-air-conditioning systems, and automobiles.

The International Space Station utilizes a highly redundant, fault tolerant, software configurable, complex, 1553 bus system that links all major sub-systems. All sensor and monitoring information is communicated using this bus and sent to the ground station via telemetry. It is, therefore, a critical system and any failures in the bus system need to be diagnosed promptly. We have modeled a representative section of the ISS 1553 bus system using publicly accessible information. In this paper, we present our modeling and analysis results, and our Telediagnosis solution for monitoring and diagnosis of the ISS based on Telemetry data.

Keywords: RDS, Remote Diagnosis Server, 1553, fault-tolerant, redundancy, multisignal modeling, TEAMS

INTRODUCTION

Modern systems such as advanced transportation systems, nuclear power plants, and manufacturing facilities, are all examples of mission critical systems that need to be monitored around the clock. These are also highly connected network enabled systems, consisting of embedded sensors in networked subsystems that can transmit data to central (or remote) monitoring stations. Prime examples of such systems include NASA's Space Shuttle and International Space Station (ISS) systems, which rely on elaborate ground support systems for monitoring and management of system health. NASA mission control utilizes a highly trained team of engineers to provide ground support for all space missions. However, such elaborate ground support infrastructure was primarily designed to support missions of finite duration. For open-ended missions, such as that of the ISS, this is economically infeasible. A fast, scalable remote monitoring system is needed to continuously monitor the telemetry stream from the ISS, thereby reducing staffing requirements for around-the-clock monitoring. Further, this software system should be able to process the alarms, form a diagnosis, assess problem severity and its impact on mission, look up resolution procedures, and guided the engineer or astronaut through a optimized troubleshooting process, thereby improving response time to events, and providing just-in-time maintenance procedures and training to support staff.

At Qualtech Systems, we have developed a Remote Diagnosis Server (RDS) under a NASA Phase II SBIR that can support multiple simultaneous diagnostic and maintenance sessions from a variety of remote systems. Clients can connect to RDS over networks (wired, wireless, dialup connections etc.) and get health assessment and intelligent troubleshooting procedures over a web browser. The solution scales easily to hundreds of sessions in any modern workstation or server. The ISS 1553 bus system is a 3 tier complex and fault-tolerant bus system that interconnects all major systems onboard the space station. It is a critical system of the ISS that needs to be monitored around the clock. We selected this as the target system for our remote monitoring solution. Working with NASA Ames Research Center, we have developed a model of the 1553 bus system onboard the ISS. Actual application of the RDS system to monitor the 1553 bus system of the ISS based on telemetry data was considered beyond the scope of this SBIR, and will be pursued in a separate effort.

This paper summarizes our efforts at developing a comprehensive remote diagnosis solution within the scope of a Phase II SBIR. In the following section, we briefly outline our RDS solution, followed by an ISS 1553 model. We assess the fault detection and isolation capability of the proposed solution by performing testability analysis on the model, and also run simulation and scalability tests on RDS to assess its applicability to real-time monitoring and diagnosis. Finally, we outline our ongoing effort with Honeywell Space Systems to build on this effort and provide a comprehensive telediagnosis solution for the ISS.

THE REMOTE DIAGNOSIS SERVER

The RDS framework is inspired by the CORBA [1] (Common Object Request Broker Architecture) in that it allows client programs to remotely access diagnosis services over a network, and there is a central computer or broker that matches the clients to the service provider. Similar to CORBA, all data is encapsulated in "strings", or "serialized", to enable the clients to invoke RDS services. We also borrowed concepts from shared memory architecture and messaging protocols such as the Tooltalk protocol [2], to add functionality for message buffering, queuing and dispatching. In addition, we implemented concepts of "Handler" and "Observer" from the Tooltalk protocol, so that we could implement supervisory or reporting functions on top of normal monitoring and diagnosis services. Agents are used to incorporate existing or legacy services into the RDS framework with minimal modifications, thus avoiding re-engineering of proven and tested software tools. The RDS framework makes QSI Integrated Diagnostic Toolset [3] (consisting of TEAMS, TEAMATE, TEAMS-RT and TEAMS-KB) accessible over the network to any communication capable system in need of diagnosis.

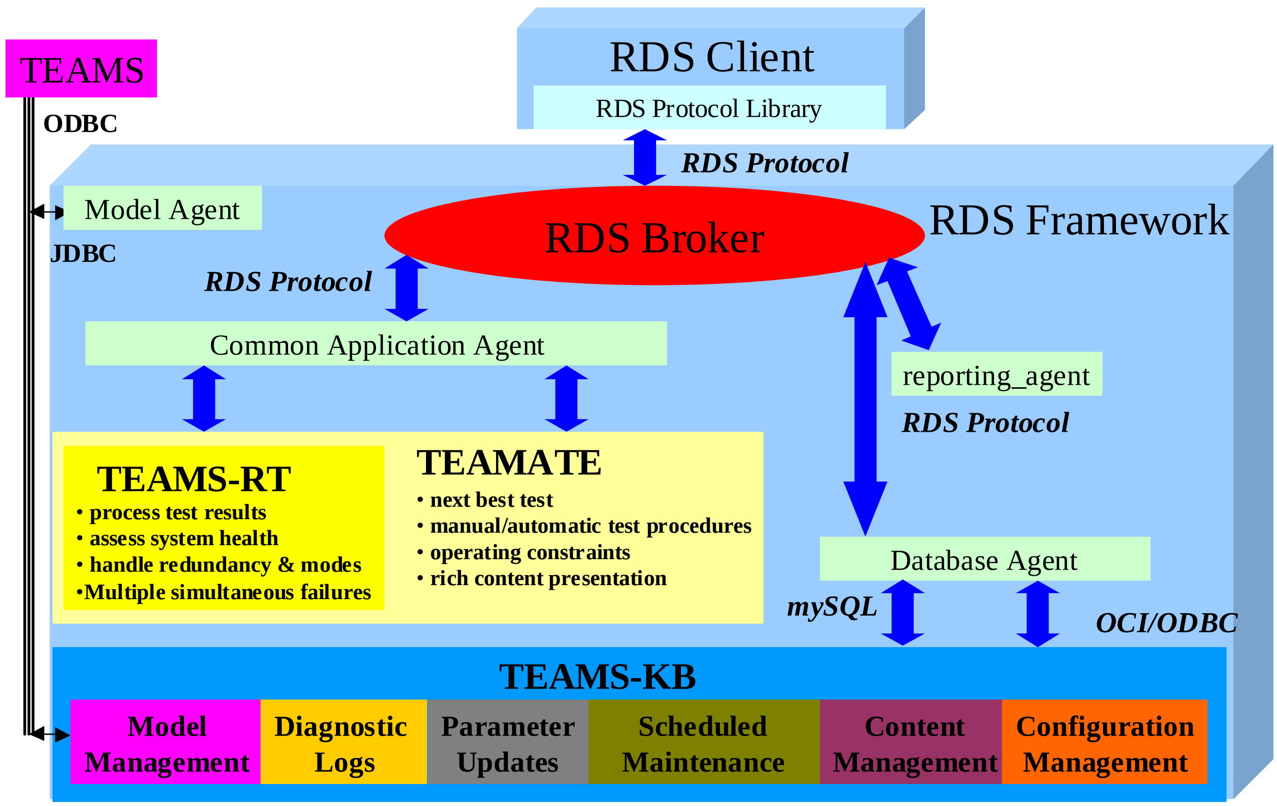

Figure 1: The Remote Diagnosis Server Framework

The RDS architecture is built around the philosophy that the scalability of the software will be derived from the Broker and, therefore, the Broker will have to be very efficient and lightweight. Further, it should be possible to add new functionality without increasing the complexity of the Broker. Consequently, the Broker is service neutral by design. While it manages the constituent services and sessions, it has no knowledge of the underlying mechanisms or data dependency of the individual services. The constituent services of the RDS are implemented by the appropriate service providers (e.g., TEAMS-RT and TEAMATE), as abstracted to the Broker by the corresponding agent. The beauty of the architecture (Figure 1) is that all tasks are performed by a multitude of agents, each with a specialized function, while the broker performs housekeeping functions, such as session and buffer management, and garbage collection. The resultant solution is also more manageable and extensible compared to alternate monolithic architectures, and scales efficiently from desktop computers to servers with dozens of processors. It also supports upwards of 300 concurrent clients in modest workgroup server configurations.

The RDS architecture is a scalable three-tier client-server architecture consisting of:

- Clients that send sensor results across the network. To facilitate development of clients that can communicate with the RDS over the network and through firewalls, we provide our customers with ANSI C implementations of the RDS protocol and transport protocols in the form of shared library objects (see Figure 2),

- A middle layer, consisting of brokers and agents, that perform session management, flow control, message buffering and routing, and load balancing, and,
- QSI's reasoner (TEAMS-RT and TEAMATE) and knowledge-base (TEAMS-KB) products at the backend.

The reasoning is model driven, utilizing multisignal models [4] developed in TEAMS. The essential constructs of the RDS framework have been described in earlier papers [5,6] and are not repeated here. More information and PDF versions of the papers are available in http://www.teamqsi.com/rds.

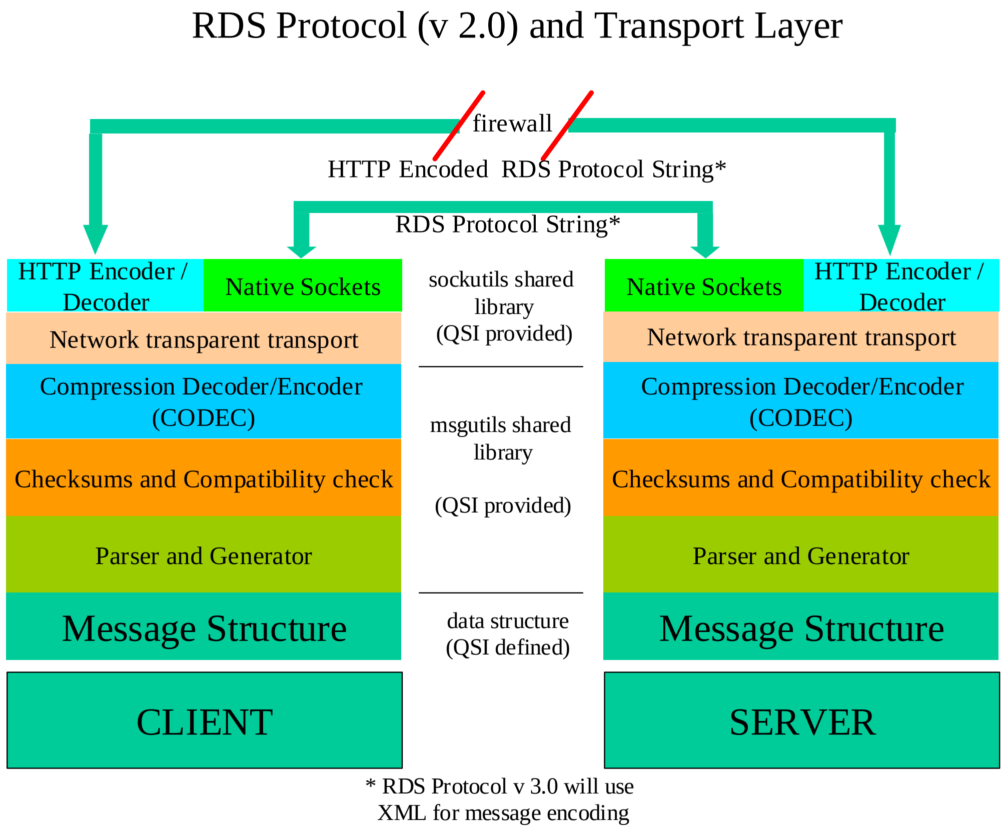

Figure 2: RDS Protocol and Transport Library – simplifying low cost remote client development.

MODELING OF THE ISS 1553 BUS SYSTEM

1. Overview

A bus system is a tricky system to model. This is because in a bus system, there is a path from each node to every other node. Therefore, certain types of failures could render the entire bus unusable. However, during normal operation of the bus, most communication is point-to-point, and only a few selected devices (e.g., bus controllers and monitors) can detect effects of failures in nodes. Further, in bus systems with multiple controllers and channels, the logical topology of the bus changes depending on the operational configuration. A key objective of this modeling effort was to retain close conformity to the structure and still be able to model the aforementioned complex and dynamic dependencies. This is because models that are close to structure are easier to peer-review and validate, and hence are easier to trust. The close conformity with structure also enables use of a single hierarchical model to satisfy the needs of multiple levels of maintenance, and call out the appropriate replaceable unit (LRU, SRU, component etc.) consistent with the maintenance level. Such diagnostic models of bus system, with close conformity to structure, are only possible with multisignal modeling [4], implemented in QSI's integrated diagnosis toolset [3]. Further, with test levels, operations, and resources assigned to tests, the diagnostic strategy can be dynamically tuned to different levels of instrumentation.

The International Space Station is a highly redundant fault-tolerant system. Redundancies are modeled directly via AND nodes in multisignal modeling methodology [4]. We could therefore model the redundant buses, channels, and controllers without having to explicitly enumerate all relevant multiple failure combinations. The ISS implementation of the 1553 bus system [7] includes multiple redundant buses, each with dual channels and controllers, and multiple redundant Multiplexer/Demultiplexer (MDM) at each level of hierarchy. We used the "SWITCH" construct of multisignal modeling methodology [4] to model various possible operational states of the ISS 1553 bus system. Thus, while the ISS 1553 bus systems can literally operate in hundreds of different combinations of MDMs and buses and controllers, the multisignal model of the 1553 bus presented here is a unified model for all such operational modes of buses and controllers.

In the following subsections, we provide a top down description of the ISS 1553 bus system model. The entire model is made out of a handful of basic building blocks, such as the transmitter, the receiver and the bus couplers. Additional information regarding the modeling of such low-level components is provided in the 1553 Modeling report [8] submitted to NASA Ames Research Center. In this paper we focus on modeling of the high-level features of the 1553 bus system.

2. ISS 1553 bus architecture overview

The ISS 1553 bus system consists of a 3 tier Command and Data Handling (CDH) System [7] (Figure 3). Each of the interface computers receive their telemetry from and send their commands to the CDH computers. The CDH System consists of 25 processing computers interconnected by data buses that collect, process, and distribute both data and commands. The CDH computers exchange data and commands in a hierarchical functional structure referred to as "tiers." This is implemented by grouping the computers and associated data buses into three tiers called the control tier, the local tier, and the user tier. Figure 3 signifies that the highest tier, the control tier, has the fewest number of computers, while the lowest tier, the user tier, has the greatest amount of computers. (The tiers are often referred to by number - Tier 1, Tier 2, and Tier 3 - rather than by their functional name.)

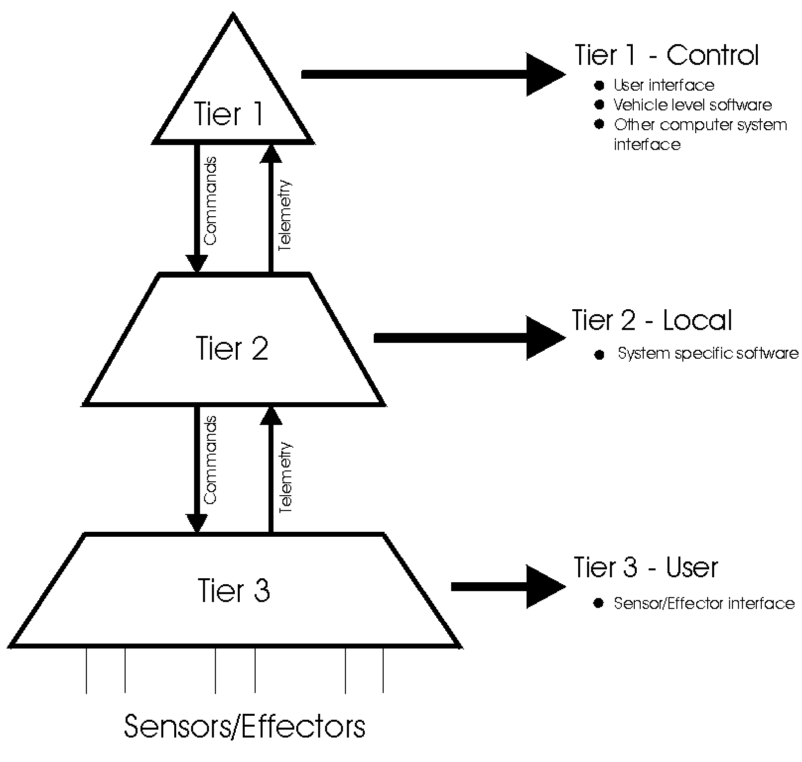

Figure 3: Conceptual view of CDH architecture

A key operational consideration to this tiered architecture is the flow of commands and telemetry. As depicted in Figure 3, for commands to reach an effector attached to a Tier 3 computer, they must start at Tier 1, pass through Tier 2 and on to Tier 3. Conversely, data (telemetry) from sensors attached to Tier 3 computers must go from Tier 3 to Tier 2 to Tier 1. Crews and controllers are only able to access data that has been passed all the way to the Tier I computers.

The control tier, as the name implies, provides the interface for the crew and the controllers. The primary purpose of Tier 2 is to execute system-specific application software. An example of this Tier 2 application software includes Guidance, Navigation, and Control (GNC) software that converts Control Moment Gyroscope (CMG) gimbal angles and gimbal rates into momentum states. Tier 2 computers are connected via 1553 data buses to the Tier 3 computers. The Tier 3 computers provide input/output processing to the thousands of sensors and effectors on the Station. Examples of sensors and effectors that Tier 3 computers interface to include temperature sensors, pressure sensors, rack flow control assemblies, and Remote Power Controllers (RPCs). The Tier 3 computers complete such processing as converting the sensor analog data to digital data and monitoring the condition of the attached hardware. Thus, Tiers 1, 2, and 3 provide the crew/controller interface, execution of system application software, and sensor/effector interface respectively.

3. Fault-tolerance in the 3-tier ISS 1553 bus architecture

An aspect of the tiered architecture is the redundancy scheme. Generally, the Tier I computers are two fault tolerant (three identical computers); the Tier 2 computers are one fault tolerant (two identical computers); and the Tier 3 computers are zero fault tolerant (only one computer with that specific set of software). However, some redundancy in Tier 3 computers is obtained by a complex allocation of software between computers. This redundancy may be obtained by tying redundant strings of sensors and effectors to the different Tier 3 computers or in some cases, placing software that performs some redundant functions in the Tier 3 computers.

Figure 4 shows a functional layout of the tiered architecture and redundancy of MDMs at a 5A configuration. 'The Tier I MDMs are located at the top of the schematic while the Tier 3s are located toward the bottom. The schematic also shows the bus connectivity of the MDMs. Looking at Figure 4 in more detail, we can see specific examples of redundancy. There are three identical Tier I MDMs, called the Command and Control (C&C) MDMs. The nomenclature for MDMs identifies the primary function of the MDM followed by an indicator for the instance of the MDM. For example: C&C-2 or C&C-3 are redundant C&C computers to the C&C- I MDM. One of the C&C MDMs is fully operational, while a second is a "warm" backup (powered on and processing data but not commanding equipment) and the third is a “cold" backup (powered off). There are five pairs of Tier 2 MDMS; each MDM in the pair is identical to the other MDM. Typically, one MDM is operational and the second of the pairs powered off. However, the redundant GNC MDM is a warm backup. There are 12 Tier 3 MDMS. None of them are exactly alike, but MDMs performing similar functions are labeled similarly. For example: LA- 1, LA-2, and LA-3. All Tier 3 MDMs are nominally powered on and operational.

The trapezoidal boxes in Figure 4 represent MDMs. These computers do not just complete multiplexing and demultiplexing tasks , they run application software and process information. The MDMs exchange data and commands between themselves via 1553B buses. These are shown in Figure 4 as vertical and horizontal lines. They are referred to as 1553B buses because they adhere to the bus protocol established in the Military Standard Number 1553B. While Figure 4 only shows the buses used between MDMs and other key computer system components, 1553B buses are also used on the ISS for communication between a CDH MDM and "smart" components in other, non-CDH Systems. These are depicted as bus "stubs" on the drawing. Smart components are those which have the ability to process their own information, such as firmware controllers.

4. Features of the 1553B bus system

A 1553B bus consists of two twisted, shielded pairs of copper wires. For the ISS, each 1553B bus consists of two channels, each channel consists of a pair of copper wires. The two channels provide redundancy, but only one channel is active at a time. If one channel fails, the other is available to take over communications. Channel changeover is supposed to occur with minimal impact to operations. Typically, the two channels of a bus are physically routed separately within a module to enhance redundancy. For example, Channel A is in one standoff, Channel B is in another. However, they are routed together through the bulkhead.

Communication occurs on buses in one direction at a time, and it must be precisely timed to prevent collisions. The speed of the bus is quite slow, 1 Megabits/second (as compared to fiber optic networks, that operate at approximately 100 Megabits/second), but it follows the Military Standard 1553B protocol. Although speed is sacrificed by using this protocol, there are several positive reasons for using the 1553B bus. Specifically, the 1553B is well-proven in space. Additionally, it has significant built-in redundancy capabilities that make it a good choice for space applications.

The bus naming convention used in the CDH System and represented in Figure 4 is as follows: there are three parts to the bus name; the first part indicates the tier of the bus, CB for control bus (Tier 1), LB for local bus (Tier 2), and UB for user bus (Tier 3). This is followed by the connectivity below it, such as INT for the internal MDM or EPS for electrical components.

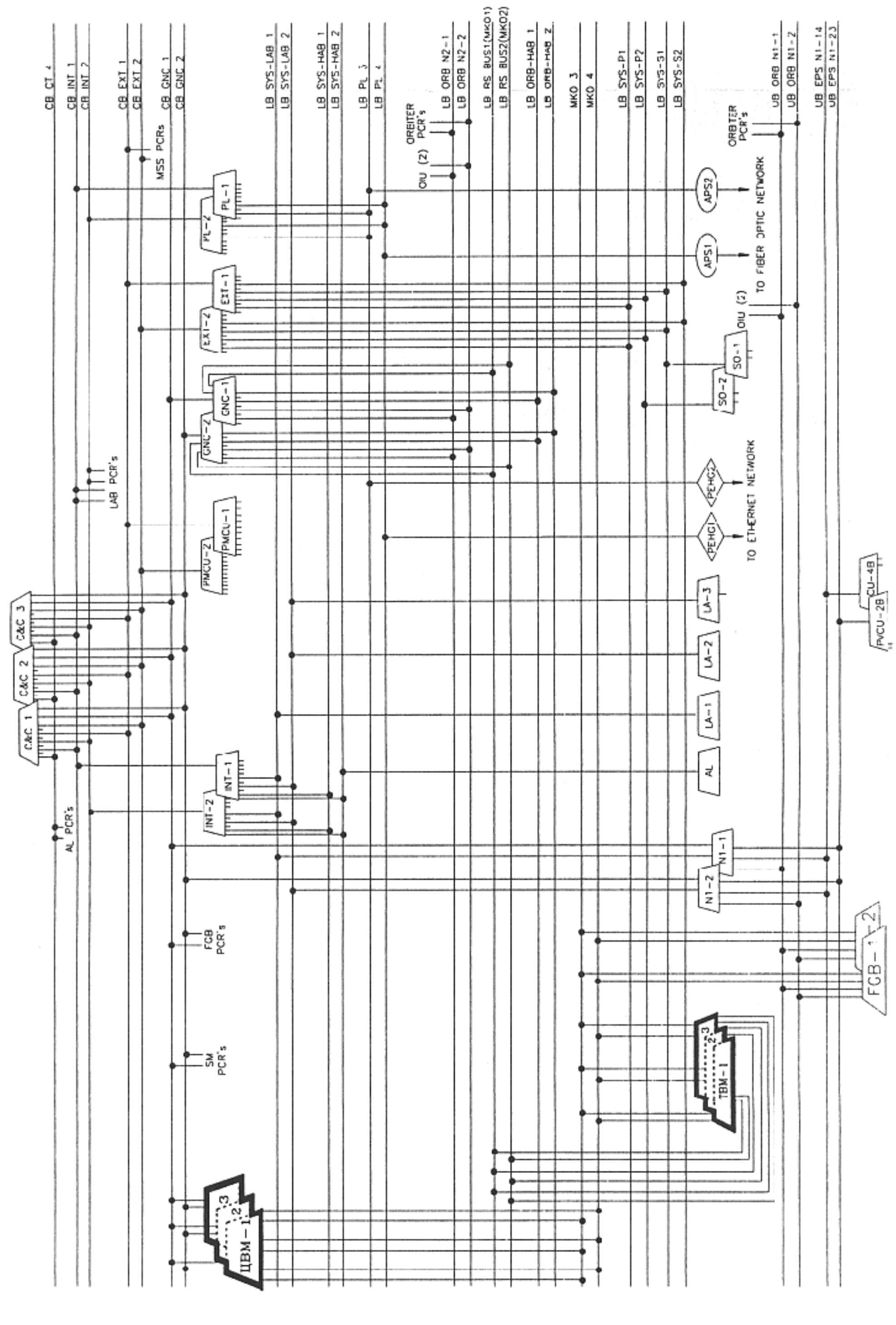

Figure 4: ISS 1553 Bus Architecture

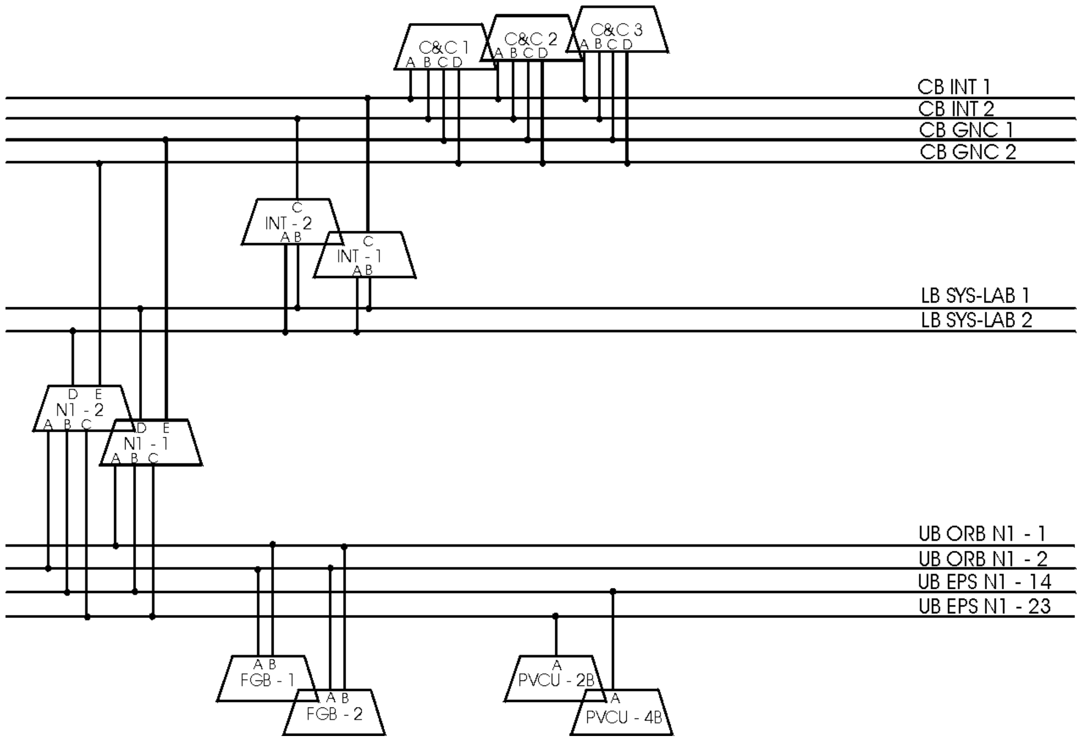

Figure 5: Subset of ISS 1553 tiered architecture selected for the current modeling effort

5. Model of a subsection of ISS 1553 bus system

Figure 5 presents the top-level representation of a portion of the ISS 1553 bus that was modeled in the SBIR effort. Figure 6 denotes the equivalent model as represented in TEAMS [7]. The five Multiplexers/Demultiplexers (MDMs) are the grey colored modules placed down the center of the work area and include C_and_C, INT, N1, FGB, and PVCU. The red modules to the left of the MDMs represent the ten 1553 "A" buses. Being a dual bus system, the green modules representing the ten 1553 "B" buses are placed to the right of the MDMs. The links connecting the buses to the MDMs are also colored red and green so as to differentiate between the "A" and "B" bus modes. The links are drawn to connect the MDMs to the buses as per Figure 5. The input (left) pins on the MDMs represent the "transmit" mode. The adjacent green and red links at the input and output of the MDMs represent the redundant buses. While the TEAMS model (Figure 6) look quite different from the schematic depicted in Figure 5, there is a one-to-one correspondence in the structural elements. Further, as is evident from the symmetry of the model, it is extremely easy to add new devices to the bus. A detailed report on modeling of the 1553 bus system is available in [8], and not repeated here. Instead, we present excerpts from the models of the MDM and multiple controllers in the following subsection.

Figure 7 presents a screendump of the first MDM (C_and_C). The three controllers within the MDM are modeled as separate modules. Switches have been placed at the inputs and outputs of the controller to represent the three different modes of operation (primary, secondary, and off line) for each controller. When controller C_and_C1 is in "primary" mode, the switches on the input and output pins are closed, enabling the controller to both transmit and receive. In "secondary" mode, only the input pins are connected to the bus, enabling the controller to receive (listen) only. In "off line" mode the switches on the input and output pins are open, thus completely disabling the controller. The controller modules are capable of decoding all the status words, and hence can assess the health of the system. These are modeled as software tests. Two switches are added to reflect the fact that these devices can be instructed to go off-line via software. Thus, these switches will open up, when instructed to do so. For example, the default states of the switches are down (i.e., connected), but they will open up if the system is in CONTROLLER-2-ONLY mode.

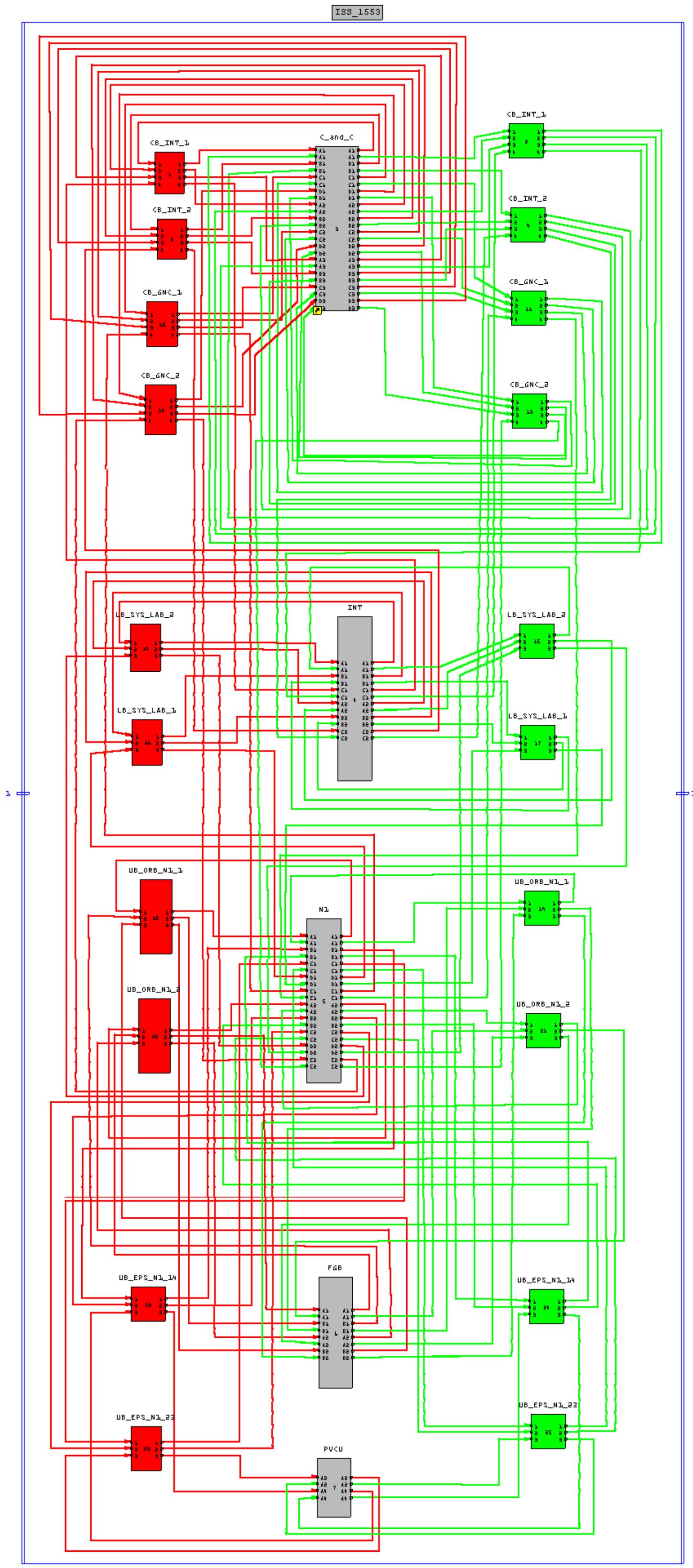

Figure 6: Top Level Model of ISS 1553 Tiered Architecture

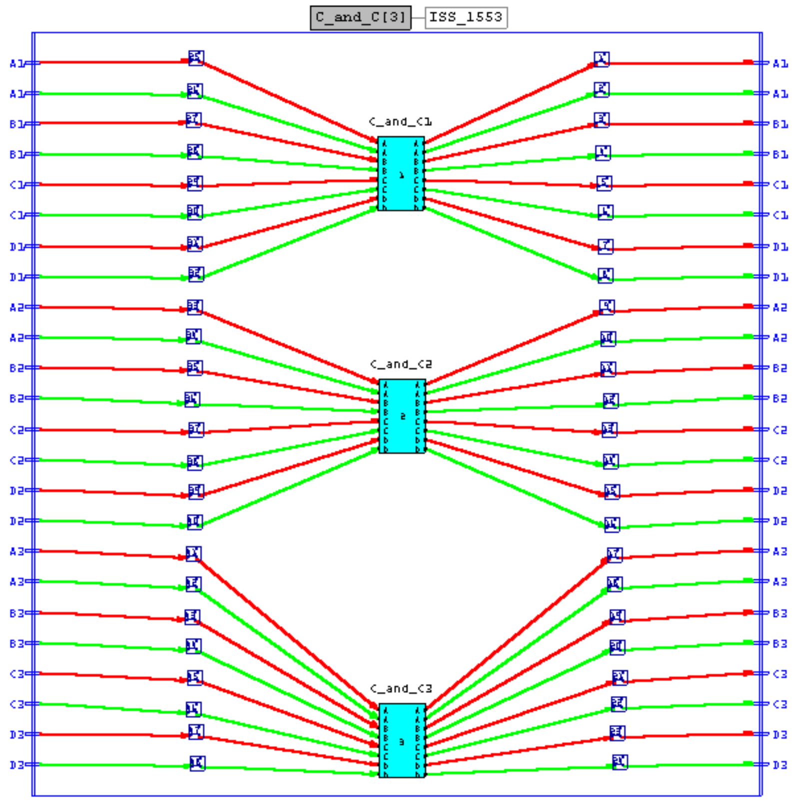

Figure 7: MDM Model

ANALYSIS RESULTS

Testability analysis enables us to assess the inherent testability of the ISS 1553 bus system and quantify the effectiveness of the proposed remote monitoring solution in terms of its fault detection and isolation capabilities. We performed testability analysis on the model utilizing the various options offered by TEAMS consistent with the various modes of operation to which the ISS 1553 bus system will be subjected. For example, if the 1553 bus system were to be monitored based solely on the software tests accessible in the telemetry stream, less that 15% of the faults can be isolated uniquely, although most of the faults would be detectable (Figure 8). However, the fault isolation can be improved dramatically by exercising the different paths of the bus system and using multiple controller-MDM configurations. Further, for remote monitoring applications, it is only necessary to isolate the failure to a LRU. Under such conditions, 95% fault isolation is achievable (see Figure 9). Thus, a remote monitoring solution utilizing our ISS 1553 model and RDS software will result in a major breakthrough in automated ground based monitoring of the space station.

We ran extensive simulations to test the performance of RDS software. Two sets of tests were performed. In the first set, we seeded random faults in our sensor_agent test data generation program [5,6], and generated data that emulates the observed test data from the telemetry stream. We then uploaded this test data to RDS for diagnosis, and compared the RDS generated diagnosis against the seeded faults. In the second set, we set up about ten concurrent test cases where the data was continually uploaded every 1 second interval, and the seeded fault randomly changed at irregular intervals. We evaluated the diagnosis accuracy on both cases, and found it closely matches the detection and isolation performance predicted by testability analysis. The second test case also measures the throughput performance of the RDS server. On a SUN workgroup server (E250) with two 400 MHz UltraSPARC II processors, the CPU utilization was under 10% during test set 2, indicating this configuration could monitor approximately a hundred similar systems onboard the ISS without any difficulty. We verified this claim by running 100 concurrent interactive diagnosis sessions with RDS and TEAMATE without any computational bottleneck. Further, the processor and memory utilization scaled almost linearly with the number of sessions, while the response time remained below the 1-second data interval. Based on these results, and simulation tests on many other models, we feel confident that our RDS solution can perform effective concurrent monitoring and diagnosis of tens and hundreds of ISS systems in real time.

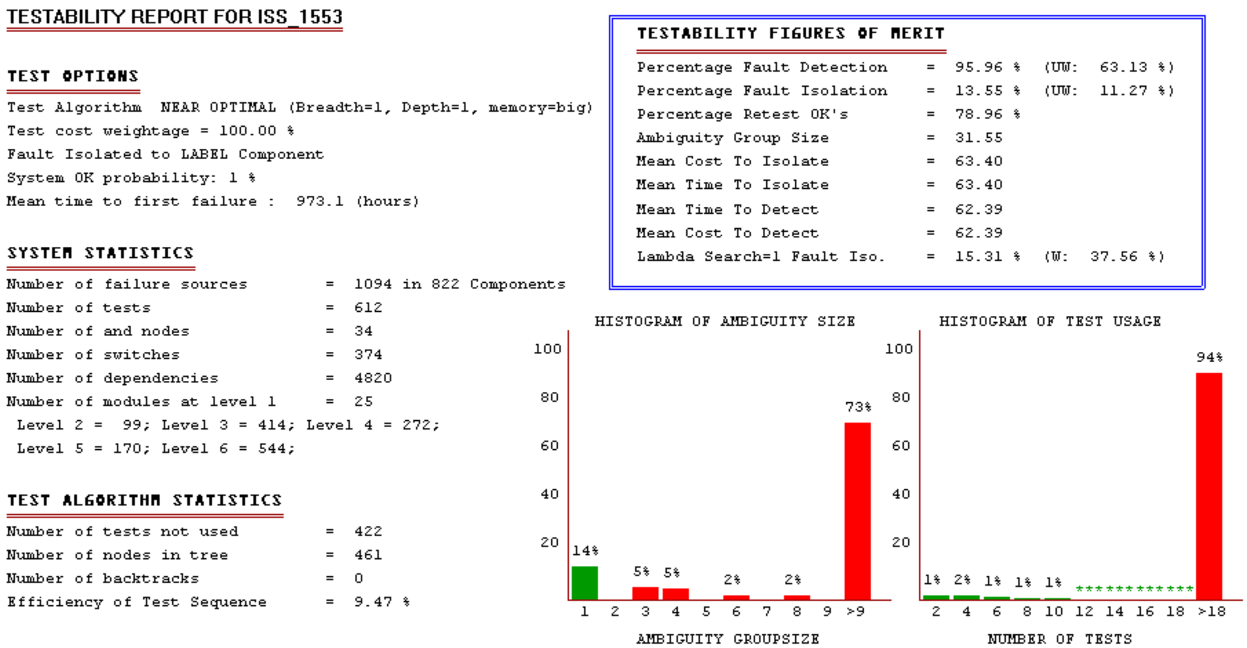

Figure 8: TFOM for isolation to component level using only status words received by the controllers.

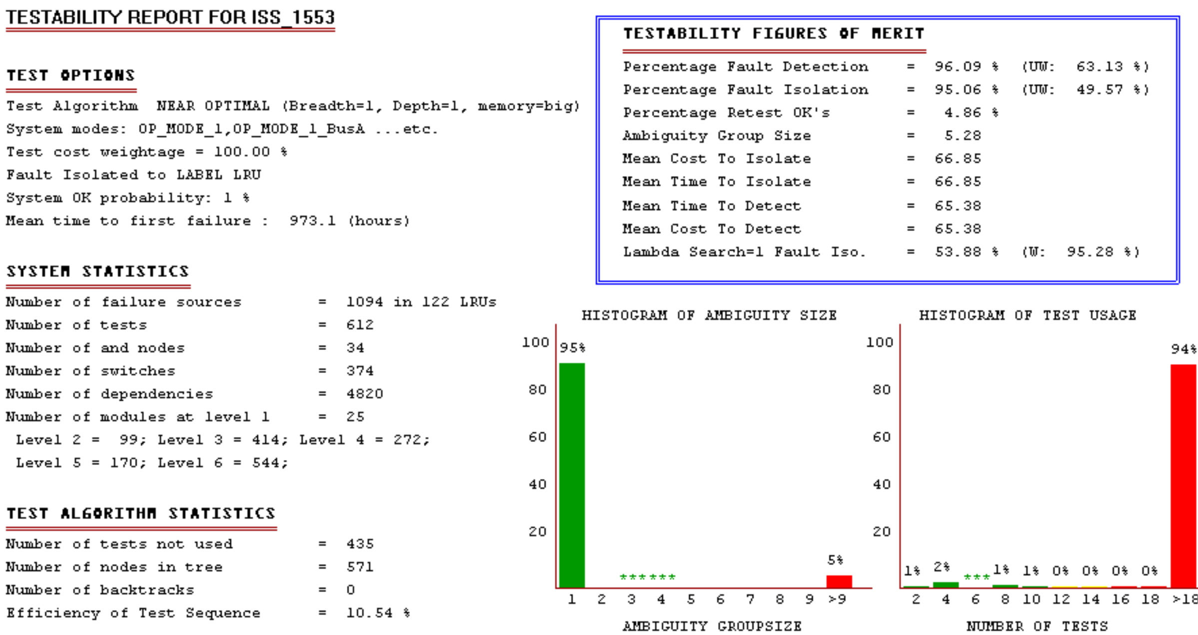

Figure 9: TFOM for isolation to LRUs using software tests only as is expected in-flight using multiple modes.

ONGOING EFFORTS IN TELE-DIAGNOSIS

Our current Phase II SBIR effort will conclude in March of 2001. However, there is considerable interest from NASA and other leading aerospace companies that lead us to believe that the RDS solution will soon be applied to the dignosis of remote systems based on field data. For example, Honeywell is using our RDS software and TEAMS models to develop a comprehensive solution for remote monitoring of the space station based on telemetry data. In an internal project, they are setting up software to mimic real-time extraction of sensor and test data from the telemetry stream, and using RDS for health assessment and diagnosis. For the demonstration, Honeywell has selected a portion of the power distribution system identified as LAAFT-2B Power Distribution Assembly. Figure 10 depicts the top-level model of this system. Primary voltage is fed through an Integrated Diode Assembly (IDA) to the DC to DC Converter Unit (DDCU). This steps down the voltage to the user-required regulated voltage level. The secondary voltage provided by the DDCU is then sent to the Secondary Power Distribution Assembly (SPDA). The SPDA is comprised of eight Remote Power Controller Modules (RPCMs). Each of the RPCMs will further distribute the power to the individual loads via the Remote Power Controllers.

The various subsystems in the LAAFT-2B are controlled and monitored through a 1553 bus. Sensors throughout the DDCU monitor voltage, current and temperature. These signals are sent to a central control over the 1553 bus. The status of the DDCU bus controller is monitored and sent to a central control. The individual RPC outputs are monitored and can be switched on and off based on the status of the monitored signals. For example, if the output of an individual channel of an RPC is drawing too much current, it can be shut down. The status of the RPCM bus controller is also monitored and sent to a central control.

The redundant MDMs comprising the 1553 bus system utilized for this selected portion of the power distribution system have also been modeled. Modeling of the redundant MDMs is accomplished with various switches. The switch states are grouped into system modes. Preliminary analysis indicates that it will be possible to achieve fault isolation down to two or less replaceable items.

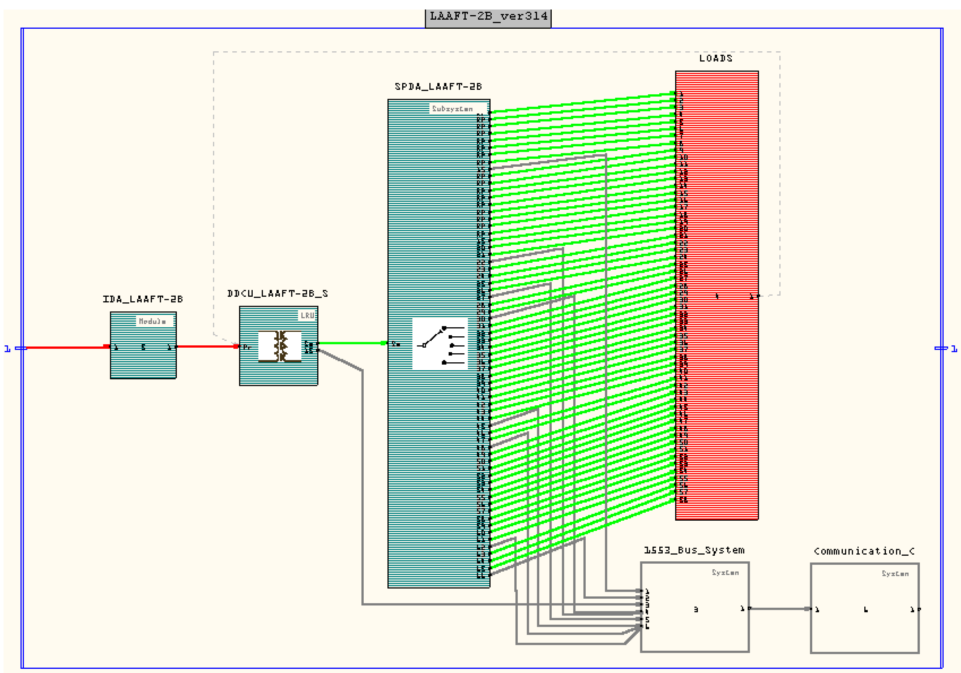

Figure 10: LAAFT-2B System Model

CONCLUSIONS

In this paper, we presented our experience in developing a tele-diagnosis solution for monitoring and health management of the International Space Station. The solution involves a scalable Remote Diagnosis Server (RDS) that employs model-based reasoning tools to assess system health and guide users through interactive troubleshooting procedures. We also developed models for the 1553 bus systems of the ISS, a highly redundant, re-configurable, dynamic, fault-tolerant system. RDS can simultaneously support hundreds of client sessions over standard Internet protocols such as TCP/IP and http. Thus, the application of RDS need not be confined to the ISS alone.

In fact, the development of RDS is a major milestone in our plan for commercializing integrated system design, diagnostic and prognostic tools. Our integrated toolset helps achieve lower life-cycle costs by addressing reliability, testability and maintainability issues: failure analysis, design for testability, automated testing, interactive diagnosis, and real-time system health monitoring. While many of our competitors offer products in the areas of integrated diagnosis, most lack a real-time diagnosis engine, and none have a networked diagnosis server capability. Until now, real-time diagnosis and prognosis have been available to a selected few multi-million dollar applications. The remarkable aspect of this technology is that it is accessible over Internet and modems, making real-time diagnosis universally accessible! This is a key discriminating factor that will enable us to reach beyond the niche market of integrated diagnosis, and tap into consumer applications and e-business.

For example, the modern automobile has enough sensors to detect the slightest performance problem. The engine computer(s) monitor fuel mixture and ignition system for optimal fuel efficiency, drive-train computer(s) monitor the grade of the road, torque and acceleration to select the correct gear, and antilock brake systems detect wheel lock ups and dynamically adjusting for brake wear. Some high-end models already come equipped with communication links (e.g., OnStarTM by Cadillac) that can report mishaps, e.g., an accident causing airbag deployment, to a central monitoring station. In a few years, such features will be available on all cars. Presently, such communication links are offered primarily as a safety net, or as a link to customer and concierge services. However, they can easily be adapted to transmit onboard data to a RDS service where car troubles can be quickly diagnosed. It is therefore conceivable that soon, the driver of a stalled car will be able to get a prompt diagnosis using RDS service, and AAA would dispatch roadside assistance with the exact spare part required to fix the problem. The applications of RDS are not limited to the automobile. Remote health monitoring of home-care patients and battlefield soldiers are two of the more promising applications. Modern high rise buildings consist of elevators, escalators, heating and ventilation systems etc. that also need to be monitored around the clock. Utilizing RDS, a central facility could monitor entire cities of high-rise buildings from one central location.

RDS is an essential piece of technology that makes such applications feasible.

ACKNOWLEDGMENTS

This research has been sponsored by a Phase II SBIR award from NASA Ames Research Center (NAS2-99049).

REFERENCES

1. CORBA Fundamentals and Programming, Ed. Jon Siegel, John Wiley & Sons, N.Y., 1996. Also see http://www.omg.org/corba.
2. Common Desktop Environment 1.0 Tooltalk Messaging Overview (Common Desktop Environment 1.0), CDE Documentation Group, Addison-Wesley Pub Company, N.Y., 1995.
3. S. Deb et al 1997, QSI's Integrated Toolset, Proc. IEEE Autotestcon, Anaheim, CA, pp. 408-421.
4. S. Deb et. al. "Multi-Signal Flow Graphs: A novel Approach for System Testability Analysis and Fault Diagnosis," in Proc. IEEE AUTOTESTCON, Anaheim, CA, pp. 361-373, Sept. 1994.
5. S. Deb et al., 2000, Tele-diagnosis: Remote monitoring of large-scale systems, Proceedings of the IEEE Aerospace Conference, Big Sky, MT.
6. S. Deb et al., "Remote Diagnosis Server", in Proceedings of the IEEE Digital Avionics Systems Conference, Philadelphia, PA, Oct. 7-13, 2000
7. "Command and Data Handling Training Manual", International Space Station, CDH TM 21109, NASA, Mission Operations Directorate, Space Flight Training Division, Feb. 1999.
8. S. Deb, C. Domagala, "Modeling Methodology of a 1553 Bus System on the International Space Station (ISS) using TEAMS 5.5" Report submitted to NASA ARC, 9/00.