Evaluating COTS Standards for Design of Dependable Systems *

C.J. Walter
WW Technology Group
Columbia, MD
cwalter@wwtechnology.com

N. Suri
Chalmers Univ.
Gothenburg, Sweden
suri@ce.chalmers.se

T. Monaghan
US Intl. Trade Org.
Washington, D.C

Abstract

This experience report presents a study on the fault tolerance (FT) support capabilities of various COTS standards prior to their inclusion in design of dependable systems. A standalone analysis and relative comparison of the FT attributes for SCI, ATM, Futurebus+ and Fiber Channel is presented.

1 Introduction & Objectives

Recent military programs specifically target reducing the cost of design, acquisition and upgrade of systems through the use of commercial off-the-shelf (COTS) and open source standards. In order to enable systems to take advantage of commercial interface standards, these programs are actively participating with industry organizations responsible for these standards, to include requirements critical to system performance such as real-time (RT) and fault tolerance (FT). This report provides an overview of work we have performed in support of these efforts in analyzing the suitability of selected standards to real systems and applications.

The increasing use of different standards in critical applications and necessity of correct and continuous operation of system services require an in-depth analysis and understanding of the dependability features of these standards. Thus, four specific communication standards were examined to disseminate their FT capabilities and features, in the context of application needs of backplane, processing and I/O. The standards discussed in this study are: Asynchronous Transfer Mode (ATM), Fiber Channel (FC), Scalable Coherent Interface (SCI), and Futurebus+ (FB+).

Within the scope of potential military applications, different domains warrant considerably different requirements of mission times, availability, reliability, maintainability, latency, size and environmental factors. This report focuses on air applications, and on three distinct sub-categories for communications for avionics applications: backplane, extended interconnects, and sensor/video networks - their basic requirements are summarized in Table 1. The numbers in brackets indicate desired scalability ranges.

	Backplane (max)	Extended Interconnect	Senor/Video Network
# of nodes	30 (1000)	4-6 (30-40)	32(256)
Max. size	1m	30-50m	150m
Data Thruput	1.6-12 Gbs	1.6-8 Gbs	1.6-12 Gbs
FT	No SPF[a]	No SPF	No SPF
FDIR	Yes	Yes	Yes
Max. BER Bit Error Rate	10E-10 detected 10E-14 undetected	10E-10 detected	10E-10 detected

[a] Given design specs: No single point of failure shall bring down the entire box, cause dissemination of bad data, or cause loss of comm. between any two nodes.

Table 1: Summary of Avionics Requirements

2 Stand-alone Analysis of Standards

In this section, the fault handling capabilities of standards are assessed considering them as individual functional units, without reference to the overall system framework. In subsequent analysis, we address the role a specific standard is expected to perform within a systems context - these issues are deferred here on account of space constraints. The discussion in this section highlights the salient FT attributes of each standard as identified during extensive reviews.

An important *caveat* is a strongly expressed caution that appealing FT attributes (or lack of them) in any of the standards are very dependent on the functionality a standard actually performs within the context of a system. Also, the relative likelihood (and relative impact on system ops. for each application scenario) of each possible fault case in discrete standards must also be taken into consideration prior to ascertaining the overall dependability capability of each standard.

*Supported in part by NAWC N62269-95-C-0130 and NSF Career CCR 9896321

For each standard, we present (a) its functional attributes and (b) basic assessment of their FT support capabilities. Our review procedure is as follows:

1. Identify the operational/functional capabilities of each standard for covering varied fault behavior.
2. Identify weaknesses in each standard that are better handled by the other units of the system. (For example, if a standard is capable of supporting retry, but does not possess the capability of logging/storing the system state to recover from).
3. Perform relative classification of standards to identify strengths and weaknesses for FT support.

3 SCI – Capabilities & Perspectives

SCI consists of high-speed point-to-point unidirectional communication links between neighboring nodes that allow for higher bandwidths over greater distance between nodes than possible for standard high performance backplane buses. In order to provide bi-directional communications between SCI nodes, the nodes and the point-to-point unidirectional communication links must form a ring topology. Typically, the number of nodes on a ring is small and the sub-network is referred to as a ringlet. One node on each SCI ringlet is assigned housekeeping tasks (initialization, timer maintenance, discarding damaged packets) and termed as the *scrubber*. A register insertion technique allows multiple nodes to transmit simultaneously on the ring, increasing the performance as a factor of N, where N is the number of nodes on the ring. On account of the ring topology employed by SCI, the magnitude increase depends upon the specific data flow pattern of the particular application.

The approach for SCI is to define an interconnect system that scales well with increasing number of processors, provides a coherent memory system, and defines a simple interface between modules. Issues of average response time, average throughput, and fairness in order to provide the necessary features for time-shared applications are specifically part of the standard. SCI uses a packet switching protocol with a 64-bit addressing mechanism.

SCI is designed to support both message passing and shared memory paradigms. Message passing is supported without any special low level protocol support. Shared memory is supported via the cache coherency protocols and remote transaction protocols (which allow locked memory transactions to be efficiently supported with packet protocols). The coherence protocols are based on single responder directed transactions and distributed directories, where processors sharing cache lines are linked together by pointers.

The SCI/Real-Time Working Group has proposed features for priority-based preemptive arbitration and queuing protocols; support for both priority-based shared memory and message passing architectures; a standard global clock synchronization method; and a standard event notification method. In addition, FT features are provided for single-bit hard error detection and correction; and hardware sub-action fault-retry protocol support.

In the following section, we highlight possible FT strengths indicated by (+) and weaknesses (-) of the SCI standard. ± indicates attributes without a definitive perceived strength or weakness. **Note:** We utilize this nomenclature of +, - and ± across all standards.

3.1 Strengths

+ The use of ringlets provide (a) a compartmentalized fault containment regions for the system, and (b) fault isolation and associated distributed fault recovery capabilities. The basic ring configuration of SCI allows for flexibility of topologies (switches, meshes, processor grids, butterfly ringlets, redundant ringlets) - all of which facilitate scalability and FT reconfiguration. The usage of ring-wraps and ring-replace techniques can facilitate link fault recovery.

+ Fault traceability by explicit HW logging & replay mechanisms. The Command and Status Registers (CSRs) provide error logging counters.

+ Live insertion capability is a useful feature for introducing test conditions into the standard.

+ Distributed recovery list approach in the cache coherency protocol provides resiliency to any single memory fault effects.

+ SCI utilizes 16 bit polynomial CRC error check versus 32 bit CRC in ATM/FC. This may appear restrictive, however, when one considers the transfer of large contiguous data streams, the SCI mechanisms will use multiple blocks to facilitate this data transfer with 16 bit CRC coverage on *each* data block as compared to the ATM/FC use of a single 32 bit CRC on the entire data block.

3.2 Cautions

± The use of timeouts in the protocol can provide a useful detection mechanism. However, current SCI versions do not have mechanisms to support complete SW level end-to-end retry. HW retry is provided, though only at the ringlet level.

3.3 Concerns

- The priority and fairness access criteria can interfere with the correct operation of the system resulting in operational bottlenecks.

- The use of a single scrubber on each ringlet presents a risk of unit failure if the scrubber fails (The same risk applies to the of the master clock, clock and flag line). However, the RT extensions to SCI contain provisions for an alternate scrubber. Nevertheless, errors are restricted to SCI ringlets illustrating fault isolation features of SCI.
- To account for faulty CRC calculations on a receiver (as to a corrupted frame), the CRC stomp mechanism must be deferred, with stations marking the packets.
- Systems based on multiple ringlets will require and mechanisms not discussed in the standard. These must be analyzed to guarantee that recovery from failures does not result in blocking.
- The time-of-death mechanism to control packets is heavily based on the system time synchronization capabilities. Correspondingly, the CSR global time capabilities should be to determine its ability to tolerate errant clocks.
- The overall fault management, identified as a potential requirement, is not fully addressed within the standard, and would need to be provided as an additional processing layer. However, the operational protocols of the standard do represent the basic FT capabilities as well as provision for interface hooks to the higher system layers.

4 Fiber Channel (FC) – Perspectives

The ANSI FC standard is a universal interface for data channels which is optimized for the predictable transfer of large blocks of data such as those used in file transfers, disk and tape storage systems, communications, and imaging devices. FC provides bi-directional connections and support for connected and connectionless operations with separate fibers for each direction. FC is designed primarily to be a local area network; however, as it is switch based, it can also be used for networks ranging from backplane to wide area networks. The framing protocol supports variable-length frames; hardware disassembly/reassembly of sequences; and control of the fabric by delimiters. A small built-in command set provides configuration management and support for error recovery. Multiple classes of services are provided. Class 1 service offers dedicated connection between two ports with guaranteed ordered delivery. Class 2 service is frame switched with buffer-to-buffer flow control, guaranteed delivery, but order is not guaranteed. Class 3 service uses datagrams without guaranteed delivery or order of receipt. Class 4 provides extended link services that provide common functions for entity addressing, data specification, and generic status.

FC topologies can be point-to-point, cross-point switch, or arbitrated loop configurations. With point-to-point, fabric elements are not present and therefore fabric services are not available. When used in point-to-point applications, a data channel establishes a dedicated connection between two pieces of equipment. The standard does not define the implementation of the fabric but sets forth the following requirements. There must be: a single-level address domain; the number of ports are only restricted by the 24 bit ID field; heterogeneous fabric elements from multiple vendors should be supported FC is optimized for input and output as well as communications between nodes. FC can also provide processor-to-memory services even though it is not optimized for these services.

4.1 Strengths

+ As the fabric is not fixed, system re-configuration over errors can be tailored by the choice of topology. The fabric can be changed without requiring HW/SW changes to computers/controllers/peripherals.
+ Unlike “passive” interconnects between nodes in traditional networks, FC allows for an active fabric which can be self-managed for error recovery.
+ Can be configured to support either connection or connectionless services based on communication level error rates.
+ The 8B/10B code scheme provides stable dc signal balance (sends same number of 1s and 0s) for the receiver. This provides a good transition density for easier clock recovery and allows error checking for unrecognized codes.

4.2 Cautions

± Fault handling is highly dependent on the capabilities of the underlying fabric.

± The use of the “hunt” group can permit the fabric to select an alternate/idle port if the primary port is inaccessible/busy. However, this mechanism is unspecified in FC.

± FC error logs provided for at the adaptation layer could be useful in fault isolation, but may need to be augmented with additional system state information to be practically usable.

4.3 Concerns

- FC is geared towards transfer of large data blocks; consequently the impact of a fault may require re-transfer of an entire data block resulting in an inefficient fault recovery process and limited traceability.

- BER rate claimed “in the absence of faults” indicating a misunderstanding of fault definitions to include only permanent faults without consideration of the occurrence of transients.

5 Futurebus+ (FB+) – Perspectives

The FB+ standard defines a backplane bus that provides performance and scalability for single and multiple bus multiprocessor systems. Allocation of bus bandwidth to competing modules is provided by either centralized or distributed arbitration. Bus allocation rules are defined to address the needs of both RT (priority based), and fairness (equal opportunity access based) configurations. Two data transfer protocols are available: compelled or packet mode. The compelled mode relies on a master/slave handshake for every data beat; the packet mode uses a fixed length block of data with an embedded synchronization pulse. Bus, system and node management are done primarily via Control and Status Registers (CSRs) and a bus monarch. CSRs are areas of memory that contain information on a node's capability, configuration and operation. The bus monarch is a node that controls the configuration and initialization of the bus. The bus monarch is selected during bus initialization through a competition between capable nodes and performs a number of duties. These include: determining the node population, polling nodes for self-test status, deciding the arbitration type (central or distributed) and enabling nodes accordingly, programming arbitration priorities and propagation delays, setting parallel protocol configuration, enabling memory space, and performing extended diagnostics.

5.1 Strengths

+ The use of the geographical address parity bit and the alternating parity bit can aid the process of fault detection.
+ Most connections can be converted to split transactions in order to prevent long data latency and as alternate channels over errors.
+ The availability of the reflected field, error detection and correction for address/data lines, protocol violation monitors, and arbitration monitors can support FT services.
+ Live insertion allows on-line replacement of faulty components and supports high-availability goals.

5.2 Cautions

± CSR's present potential capabilities for error management. However, these features are not documented/formalized in the base standard.

± FB+ presents multiple profiles; designers need be careful of the capability actually available for a chosen profile.

5.3 Concerns

- The base standard has many signals not protected by fault detection mechanisms: status, capability, arbitration condition, geographical address, central arbitration, and reset lines.
- The multi-party asynchronous bus arbitration protocols present limited resolution in identifying faults within the interfaces. Furthermore, the reliance of data transfer on the centralized arbitration mechanism can result in high fault susceptibility. This aspect is typically handled by providing for a redundant arbiter; however, the functionality is still that of a centralized unit.
- The false detection of signaled acknowledgments can result in potential for faulty reads of capability modes or of transferred data. There also exists a potential for undetected bus errors, based on occurrence of transient control line errors with specific timing windows (cases of early release of bus during the time the slave is reading data). The use of a double-read of data or the longitudinal parity can possibly alleviate this problem.

6 ATM – Perspectives

ATM aims to provide a flexible facility for the transmission and switching of mixed media traffic comprising voice, video and data. It provides a multiplexing and switching method for Broadband Integrated services Digital Networks (B-ISDN) based on fixed size cells and header information which identifies explicit channel information. ATM provides high performance with low latency and high capacity based on a constant bit rate. Variable bit rate services can also be handled to allow for flexible allocation of bandwidth among users. ATM uses small packets that are transported over lightweight virtual circuits that are a fixed 53-bytes (48-byte payload, 5-byte header). This approach allows the protocol to be independent of the physical layer and application hardware. Connections can be established as either permanent or switched via these virtual circuits. Cell routing is based on a two-level addressing structure: the virtual path (VPI) and virtual channel (VCI) indicators. The end-to-end concatenation of VCIs is the virtual channel connection (VCC) which are uni-directional and do not provide for error recovery or flow control. The VPI identifies the physical path that is associated with a set of VCIs. All cells associated with a VCC are transported along the same route through the network and delivered with the cell sequence preserved.

6.1 Strengths

+ Capability for handling constant- rate, variable-rate, connectionless and connection-oriented framing, multiplexing, and transport services are useful over system re-configurations.
+ The generic flow control field defined at the user network interface is capable of implementing a modest level of congestion control.
+ Capability for negotiating quality of service parameters allows for efficient allocation of bandwidth and priority management over error handling. ATM cell related parameters include: error, loss and block ratios; mis-insertion rate, transfer delays and variations.

6.2 Cautions

± There is a high dependency (and deferred responsibility) for fault and FCR's error management on the ATM Switch Fabric, the ATM's AAL and other higher levels/controllers.

± ATM switching architectures have their own classification with related strengths and weaknesses.

6.3 Concerns

- As the ATM operates on a virtual cell/connection basis, it can be difficult to implement a fault retry capability. Also, as no error control is performed within the network, the future fault-tolerance capabilities are limited by the information encapsulation possible within the 5-byte block header.
- Similar latency/detection liabilities as FC due to the transfer of large data blocks.
- The data message header is provided with an 8-bit CRC for every 32-bits. However, data error checks occur only at the Adaptation Layer level and are handled solely by the use of a 32 bit CRC for the entire block. (See discussion of CRC use in the section on SCI)
- Limited capabilities for path re-establishment on the occurrence of faults in the initial message route. The corruption of the routing table can become a major hindrance to system operation.
- Cases with faults in the priority assignment and handling mechanism impact system performance.
- Limited fault detection and recovery capabilities based on the Alarm Indication Signals (AIS) and use of Far-End-Receive-Failure (FERF) signals; recovery limited to simple message re-transmittal.

7 Discussion & Comparative Analysis/Applications of the Standards

Based on the standalone properties of standards, Table 2 attempts to summarize the capabilities of each standard in different application scenarios. The various expected roles a standard is required to perform in a system are broadly classified in Table 2. A key basis for matching of standards and applications is the operational capabilities of the standards and the desired operational specifications. Unless this basic match is achieved, a match of a standard to any dependability attribute is of little value.

The initial intent of the analysis was to formulate a standard test-set of fault types and grade each individual standard accordingly with respect to their fault-handling capabilities. A proper comparison was impossible due to the lack of a common fault model across all the standards. It is perhaps more relevant to quantify the general number of fault scenarios possible in each standard and to document the nature of fault-tolerance capabilities or liabilities in each standard and use this for comparison across the standards.

At the design stage, the functional, performance and throughput aspects of these standards were the design drivers, and not the FT issues which only come to the fore when these standards are utilized to create dependable services. This situation results in many fault-tolerance attributes being introduced as add-on features. Thus, the basis of our evaluation of these standards is to consider:

- Which standard offers better possibilities for supporting additive fault-tolerance capabilities?
- Which standard can possibly be used without proving detrimental to the overall system level fault-tolerance capabilities (i.e., by providing hooks or the capability of deferring fault handling to a higher system layer).

The design trends appear to aim at providing a sufficient set of mechanisms so that a standard can be regarded as a "highly dependable" entity capable of performing the desired functions with a high degree of resilience to faults. This approach is acceptable, although one needs to be cautious that there is no guarantee that a dependable system will necessarily result. The dependability of a system is very much a function of the overall system paradigm, the operational structure of the system and the specific resource and redundancy management protocols used. It should be noted that the coverage of faults from a system level perspective may not directly reflect or relate to capabilities of fault coverage at the individual block level.

	Interconnects: Desired Attributes					*Supported by:*			
	IP[a]	BP/IO	S/V	IB	MM	SCI	ATM	FC	FB+
BW Allocation	High	V.High	V.High	V.High	Mixed	Yes	Yes	Yes	Yes
Arbitration Resolution						No	Yes	Yes	Yes
Isochronous	No	No	Yes	No	No	Yes	Yes	Yes	Yes
RT Support	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Block size	Small	Mixed	Large	Mixed	Mixed	Mixed	Large	Large	Mixed
Msg. passing	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes
Synchronization	Yes	Yes	No	Yes	Yes	Yes	Y/N[b]	Y/N	Yes
P→M transfer	Low	High	V.Low	High	V.High	Yes	Ltd.	Ltd.	Yes
Shared Memory	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes
Cache Support	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes
FT Mechanisms						CSR	Network	Network	CSR
Roll-forward Recovery	Yes	Yes	No	Yes	Yes	Yes	No	No	No
Roll-back Recovery	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Reconfigurable	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes
Latency (fault-free)	V.Low	V.Low	Low	Low	V.Low	10ns	.5ns	μs	ns
Topology	S/L[c]	S/L	Star	Linear	S/L	Ring/Switch	Switch	Switch	Linear

Table 2: Applications & Relative Comparisons Across Standards

[a] IP=Inter-Processor Comm., BP=Backplane Comm., SV=Sensor/Video, IB=Inter Block Comm., MM=Memory Mgmt.

[b] Only for specific configs. and classes of service

[c] Star/Linear

The "dependability" of a system is essentially a qualitative measure of a system's capability in being able to deliver the expected services with a desired or specified level of assurance. The aspects of functional behavior of the system and its dependability qualities are also strongly interlinked; thus, the continued emphasis on addressing the functional role of a standard from a system level perspective.

Our ongoing work addresses the issue: If the system is built using these standards, do these standards facilitate the system being able to achieve its desired or specified fault-tolerance and dependability goals? The method for achieving this goal consists of resolving a series of questions.

- What are the fault tolerance specifications of the entire system?
- What is the exact role/functionality the chosen standard provides in the system? Subsequent to this, what roles do individual standards perform best within the specified system framework?
- Considering the above mentioned aspects: Does the standard display sufficient:
  - current capability of supporting dependability requirements - can it provide a self contained fault-resilient block?
  - capability to support possible extensions which aid in better matching of the expected functionality and capabilities of the standard, and availability of interfaces to other system layers to defer error handling?

It is recommenced that such a structured approach be used to determine if the standard's attributes match the requirements of the system framework. It has already been emphasized that the nature of the system model is crucial in determining the role the standard can be expected to perform.

Overall, we have analyzed and documented the capabilities of the various standards for their basic ability to support FT. We have deliberately refrained from suggesting the use of one standard over another, as only when the role of a specific standard is known within the context of a system, can objective comparisons be made across the standards.

References

[1] M. Richards et al., *Rapid Prototyping of Application Specific Signal Processors*, Kluwer, ISBN 0-7923-9871-8, 1997.

[2] Navy's Next Generation Computer Resources (NGCR) Program, 1994+, http://www.faqs.org/rfcs/rfc1679.html

[3] C. Walter, et al., "Dependability Issues in the Reuse of Standard Components in Open Architectures," *AIAA Computing in Aerospace 10*, pp. 443-453, 1994.

[4] C. Walter, et al., "Dependability Framework for Critical Military Systems Using Commercial Standards," *AIAA 14th Digital Avionics Systems Conf.*, pp. 184-192, 1995.

[5] *Scalable Coherent Interface*, IEEE Press, #1596, 1992.

[6] *Futurebus+ – Logical Protocol Specification*, ISO/IEC 10857:1994(E) (ANSI/IEEE 896.1), 27 April 1994.

[7] *ATM User-Network Interface Specification*, ISBN 0-13-225863-3, 10 September 1993, ATM Forum/Prentice Hall.

[8] *Fiber Channel Arbitrated Loop (FC-AL)*, ANSI draft standard, revision 4.2, 11 March 1994.