CONGRUENT SOFTWARE, INC.

3998 Whittle Avenue
Oakland, CA 94602
(510) 531-5472
(510) 531-5472 FAX

FROM: Peter Johansson

TO: T10 SBP-2 *ad hoc* Working Group

DATE: June 9, 1997

RE: SBP-2 and 1394-1995 Transaction Errors

As noted by Pete McLean in his earlier memorandum, T10/97-185r0, there are a variety of Serial Bus transaction errors that require special handling by SBP-2 targets and initiators. The annex proposed below is informative, since there may be more than one way to successfully recover from one of these errors. The hope is that implementers of both hardware and software, initiator and target will be alerted by the discussion and choose appropriate strategies.

Annex D
(informative)

Serial Bus transaction error recovery

Inherent in the nature of Serial Bus as a split-transaction bus are transaction errors that can leave the requester and responder with different or ambiguous information. The most common instance occurs when an acknowledge packet transmitted after receipt of a request or response packet is corrupted and not observed by the sender of the primary packet;

When an acknowledge packet is missed by the sender of the primary packet, the sender does not know which of the following applies:

- The primary packet was correctly received by the destination node (and resultant side-effects in that node may have occurred); or
- The primary packet had a CRC or other error, has not been correctly received by the destination node and no state changes have occurred.

If an acknowledge is missing after transmission of a response packet, 1394-1995 prohibits retransmission of the response packet. Even in the case of a missing acknowledgement following a request packet, it may not be advisable to retry (because of side-effects associated with certain SBP-2 transactions).

A few of the more common error scenarios and the recommended error recover for each are described below.

D.1 MANAGEMENT_AGENT write request

When a management ORB is signaled to a target by means of an eight-byte block write to the target's MANAGEMENT_AGENT register and no acknowledgement is received, the initiator does not know whether or not the ORB will be fetched by the target.

In the case of a LOGIN ORB, error recovery is straightforward if the initiator waits a minimum of *login_timeout* for the return of a status block before any attempt is made to retry the login. By waiting the specified time the initiator avoids the possibility of multiple status blocks for the same ORB address.

For the other management ORB's, there is no clean solution provided by SBP-2 at present.

TO BE DETERMINED – How should the time-outs for the other management ORB's be established and communicated to the initiator?

D.2 ORB_POINTER write request

A write to the ORB_POINTER register is valid only when the initiator knows the target fetch agent to be in the RESET or SUSPENDED state. A consequence of the write is that, if successful, the target fetch agent transitions to the ACTIVE state. If no acknowledgement is received by the initiator after a write to the ORB_POINTER register, the initiator shall not retry the write unless it first ascertains the state of the target fetch agent. One suggestion is to read the AGENT_STATE register; alternatively, the initiator could write to the AGENT_RESET register before attempting any retry.

D.3 Data buffer, ORB or page table read request

If the target transmits a block read request and receives no acknowledgement, the read request may be retried immediately but care should be taken to not reuse the same transaction label as the failed request. The target shall wait a minimum of it's own SPLIT_TIMEOUT period before the transaction label may be reused in any subsequent request.

D.4 Status FIFO write request

When the target detects a missing acknowledgement after a write to an initiator's status FIFO, it shall take no error recovery actions. Any target resources allocated to the ORB should be released by the target. The initiator is expected to discover the error by means of a higher-level mechanism, such as a command time-out and to initiate appropriate error recovery. The nature of the error recovery undertaken by the initiator likely depends whether or not the target processes ORB's and returns their status in order, but in any event is beyond the scope of this description.