Accredited Standards Committee[*]
X3, Information Processing Systems

Doc. No.	X3T10/94-118R0
Date:	May 19, 1994
Project:	
Ref. Doc.:	
Reply to:	J. Lohmeyer

To: Membership of X3T10

From: L. Lamers

Subject: SAM Forwarding Letter Ballot Comments

SAM Letter Ballot Results: 49:3:0:7 = 59

Yes, with comment:	AT&T Global Information Solutions
No:	Hewlett Packard, IBM, Quantum
Did not respond:	AMD, Apple, Compaq, Harbor Electronics, Maxtor,P.E. Logic, Samsung

= =

NCR comment: “Delete annex A since it is temporary.”

HP comment:

SAM Review Comments - Revision 13

#001 (E) Section 1, Page 8, Paragraph 3

SCSI-2 is listed as X3.131-1992. I believe it is actually X3.131-1994.

#002 (E) Section 1, Page 8, Paragraph 6

SCSI -2 should be SCSI-2.

#003 (E) Section 2.1.12, Page 9, Paragraph 1

The term being defined and it’s use should be “completed command”, not “command complete”. This is consistent with terms like “aborted commands”. It also removed the ambiguity between this term and “Command Complete”.

#004 (E) Section 2.1.14, Page 9, Paragraph 1

This term should be “ended command”, not “command ended”.

#005 (E) Section 2.1.32, Page 10, Paragraph 1

[*] Operating under the procedures of The American National Standards Institute.

This term should be “protocol indication”, not “indication” in order to be consistent with terms like protocol service request, protocol service response and protocol service confirmation.

#006 (E) Section 2.1.58, Page 11, Paragraph 1

Why is the term “queue” defined? I thought we had purged the term from the document.

#007 (E) Section 2.1.61, Page 11, Paragraph 1

This term is, or should be, redundant with “protocol service request”. In general, I noticed that request, indication, response, and confirmation are all listed twice, once alone and once with the “protocol service” added in front. I don’t see the distinction between the two.

#008 (E) Section 2.1.62, Page 11, Paragraph 1

This term is redundant with confirmed protocol service.

#009 (E) Section 2.1.62, Page 11, Paragraph 1

This should be called “request-confirmation transaction”.

#010 (E) Section 2.1.72, Page 12, Paragraph 1

This term is redundant with “device server”.

#011 (E) Section 2.1.74, Page 12, Paragraph 1

Remove, “...while in transit”. There can be service delivery subsystem failure without the transaction ever being “in transit”. For example, by differential transcivers may be disabled on a parallel bus when a single ended device is plugged into the bus. The data was never sent out of the local system. I think that removing this part of the sentence removes the requirement to define what “in transit” means in all cases.

#012 (T) Section 2.1.81, Page 12, Paragraph 1

Pending task is undefined, yet it is used in this definition.

#013 (E) Section 2.1.84, Page 13, Paragraph 1

“Task aborted” should be “Aborted Task”. I also see this as a redundant definition of aborted task.

#014 (E) Section 2.1.85, Page 13, Paragraph 1

“Task completed” should be “Completed Task”. I also see this as a redundant definition of completed command.

#015 (E) Section 2.1.86, Page 13, Paragraph 1

What is an ended task? Is it an aborted task, or a completed task? Is it redundant with other definitions.

#016 (E) Section 2.1.8x, Page 13, Paragraph 1

Task Management Indication and Task Management Confirmation are not defined.

#017 (E) Section 2.1.93, Page 13, Paragraph 1

Termination Pending should have (task state) listed after the term. It is one of the states in your task state machine definition.

#018 (E) Section 2.4, Page 15, Paragraph 1

The sense key shall be either “INVALID FIELD IN CDB” or “INVALID FIELD IN PARAMETER LIST” depending upon where the invalid field is found. The statement, as it appears, would cause me to issue “INVALID FIELD IN CDB” even when there is an error in the parameter list of a command.

#019 (E) Section 2.5.1, Page 15, Paragraph All

Clean up the format of the columns. They are hard to read.

#020 (E) Section 2.6, Page 18, Paragraph 2

“...([input1][,input2]...]||” should be “...([input1][,input2]...||”.

Remove one “]” at the end of the input section.

#021 (E) Section 2.6, Page 18, Paragraph 10

The bullet item on the brackets should not be here. This is the same as the definition in the notation section (2.5.1). I assume that you could also include curly brackets in a procedute notation, but this is not included in the bullet list.

#022 (E) Section 3.1, Page 22, Paragraph 4

The bullet should be “c.” not “c)”.

#023 (I) Section 3.1, Page 22, Paragraph 5

Last sentence. Everything humanly possible should be done to eliminate the use of internal behavior as a description of how something works. Every item in this standard should use externally observable behavior to describe the implementations. If internal behavior is required, then we have not done our jobs.

Remove this sentence and all internal behavior descriptions.

#024 (E) Section 3.2, Page 23, Paragraph 1

Replace “request-response transaction” with “confirmed protocol service”.

#025 (E) Section 3.2, Page 23, Paragraph 3

“..., the request becomes pending upon reciept and complete when the response is passed...” should be, “..., the request becomes pending upon reciept and completes when the response is passed...”.

#026 (E) Section 3.2.1, Page 25, Paragraph 1

“reading or writing to the media.” should be “reading from or writing to the media.”.

#027 (E) Section 3.2.1, Page 26, Paragraph 2

First sentence should be removed. It is the definition of a task and is redundant with the definition in the definition section.

#028 (T) Section 3.5.2, Page 31, Paragraph Last

Remove this paragraph. We state everything by arrival order so mentioning that we assume in order confuses and implies a requirement on the service delivery subsystem. Removing the paragraph does not change the meaning of arrival.

#029 (E) Section 3.6, Page 32, Paragraph Fig 12

The service delivery interface is part of the device, but is described in the notation as being part of the service delivery subsystem. Either the figures need to change to reflect this, or the notation needs to move the SDI into the device rather than in the SDS.

#030 (T) Section 3.6, Page 33, Paragraph 5

The target is listed as not being able to originate task management functions. What is the ruling then on the FCP target being able to issue an abort exchange? Is it legal according to SAM for a target to detect that an error has occured and abort the task due to the detected error?

#031 (T) Section 3.6.1, Page 34, Paragraph 1

Since an initiator can have more than one initiator identifier, how does a target tell the difference between initiators? Is this going to be protocol specific, left undefined? In FCP for example, a login can tell the difference. In parallel it is more problematic since there is no way to tell if they are the same or if they are different.

#032 (T) Section 3.6.2, Page 35, Paragraph 10

There is no need for a base logical unit. It is no different than a logical unit zero. As a matter of fact they both have the same LUN.

#032 (E) Section 3.6.2.1, Page 36, Paragraph 1

The task manager also controls the tasks bsaed upon the ACA status, the tasks attributes (HEAD, ACA TAG, SIMPLE, ORDERED). It does not control them based soley upon the task management functions.

#033 (T) Section 3.6.3, Page 36, Paragraph 5

The task set definition, as written, does not allow more than one untagged task in the task set at a time. I would suggest the following notation:

Task Set = 0{Tagged} + 0{Untagged}

#034 (E) Section 3.6.3, Page 37, Paragraph 1

Remove the reference to queue. “Task Set (queue)” should be “Task Set”.

#035 (E) Section 3.6.3, Page 37, Paragraph 15 and 16

Tagged Task Address and Untagged Task Address are not very useful and cause confusion. You are implying that there is some special method used to refer to a task when using task management functions. There isn’t. You refer to it in the same manner in which you do when you create the task. Eliminate these two items from your notation.

#036 (E) Section 3.6.3, Page 38, Paragraph 4 and 5

“..protocol service request, The initiato r’s”, should be “...protocol service request, the initiator’s...”.

#037 (E) Section 3.7, Page 39, Paragraph 4

Change “...clause ?” to the correct reference.

#038 (E) Section 4, Page 42, Paragraph 1

“...sense data. returned...” should be “sense data returned...”

#039 (E) Section 4.1, Page 42, Paragraph All

The section 4.1 on page 42 is duplicated on top of page 43. Remove one of them.

#041 (T) Section 4.1, Page 43, Paragraph 3

Why is the statement allowed that says that media may be modified even if there is an invalid parameter or invalid field in a CDB? What cases would possibly allow you to modify media when the command is deemed to be incorrect?

#042 (T) Section 4.1.2, Page 44, Paragraph 2

The first sentence, as worded, says that the ACA will never be cleared. It should simply say that if the bit is a one, then the ACA shall be treated according 4.6.

#043 (T) Section 4.1.2, Page 44, Paragraph 3

Why is ACA = 0 required? If I want to build a SCSI-3 device, you are requiring that I keep the old CA baggage even if I don’t intend to operate with any SCSI-2 initiators. This is an unecessary requirement which prevents SCSI-4 (yikes!) from eliminating support for CA entirely.

#044 (E) Section 4.2, Page 46, Paragraph 2

Update the reference. “?” should be “5.6”.

#045 (T) Section 4.3.2, Page 48, Paragraph

An indication and response are required. The model you present here does not follow the confirmed services model presented earlier.

#046 (T) Section 4.3.3, Page 49, Paragraph

An indication and response are required. The model you present here does not follow the confirmed services model presented earlier.

#047 (E) Section 4.5.1, Page 51, Paragraph Last

“initiato r’s” should be “initiator’s”.

#048 (T) Section 4.5.2, Page 52, Paragraph

Considering the confusion over linked commands in the past, perhaps we should add in a statement that clarifies whether or not the linked command is an implied reservation or not. We have argued this one in committee a number of times and concluded only that it was not clear in the current documents. I suggest we try to add into the description the text to make it clear what we do in this case.

#049 (E) Section 4.6.1.1, Page 53, Paragraph Last

“mode byte” should be “control byte”.

#050 (T) Section 4.6.1.1, Page 54, Paragraph 1

“The task shall then be entered into the task set if it meets all other conditions for acceptance.” This does not convey that the task is the first one to be executed as was done in SCSI-2 CA. Stating that it is accepted is not sufficient. It must be stated that it is executed next and that it is untagged.

#051 (T) Section 4.6.1.2, Page 54, Paragraph 2

There should be no requirement for me to support the setting of the ACA bit to zero.

#052 (E) Section 4.6.4, Page 58, Paragraph 3

“initiato r’s” should be “initiator’s”.

#053 (T) Section 5, Page 60, Paragraph 12

Abort Task should not be required. It is only required if tagged

tasks are supported. Abort Task Set makes more sense to be required

than does Abort Task

#054 (E) Section 5.1, Page 61, Paragraph 6

“A response of Function Complete indicates that the task is not in the task set.” implies that there is some other response sent if the task was aborted. This should say that the function complete is sent once the task is aborted. Also, the task not being present is not an error.

#055 (E) Section 5.4, Page 63, Paragraph 1

“... in the specified task set shall be .” Oops, shall be what? Aborted.

#056 (T) Section 5.4, Page 63, Paragraph 1

“All data for all terminated tasks shall be cleared.” Two things. First, it should be aborted tasks not terminated. Second, the data cleared is the sense data. This implies that I must flush my buffers for any data that these tasks may use. This is not the intent of the statement.

#057 (T) Section 6.1, Page 67, Paragraph 5

Your current task definition does not include a task which is sending status. Current does not mean only data, it includes status or any other information transfer.

#058 (E) Section 6.2, Page 67, Paragraph 2 and 3

“...that were in the task set before the referenced task.” should be, “...that were entered into the task set before the referenced task was entered into the task set.”. Just saying before the referenced task leaves the ambiguity that they could be before the referenced task in the task set (i.e. HEAD).

Same comment for paragraph 3.

#059 (E) Section 6.2, Page 68, Paragraph 2

4.6.2 is not a correct reference. 6.2.1 is the correct reference.

#060 (E) Section 6.2, Page 68, Paragraph 3

“...or is unable to continue command execution because of task termination” does not correlate with this state. Task termination is a very orderly procedure. This implies that the task can’t be completed when it can, and is, completed in a predefined fashion.

#061 (E) Section 6.2.1, Page 68, Paragraph 3

This should also state that the ABORT TASK SET must also reference the initiator identifier. That is, an ABORT TASK SET from initiator 1 does not effect the tasks of intiator 2.

#062 (T) Section 6.2.1, Page 68, Paragraph 5

The occurance of the ACA condition with QErr set does not effect the task set. The CLEARING of the ACA condition when the QErr bit is set causes the tasks to be cleared.

#063 (I) Section 6.2.1, Page 68, Paragraph All

This list groups a lot of items into the category of abort. Many of these have no correlation to an abort as it is known today. For example, if I am reading from the media and get an Abort Task for a command not “internally active”, I just delete it and continue with the current task. If I get a reset, I blow everything away, including the read from the disk. These are very different, yet you are grouping them into the same term.

#064 (T) Section 6.3, Page 70, Paragraph All

This entire section is extrememly confusing. It is making assumptions about internal states of the device and is not a model based upon the externally observable behavior of the device. When we started work on the queuing model, we assumed that the internal states of a device were out of bounds for discussion. This model has put everything back into the internals of a device. I cannot vote to accept any model which is “device-centric” rather than “bus-centric”.

IBM comment:

Note: Comments marked with an * what I consider to be non-editorial issues, questions, etc..

1. Page 9 Section 2.1.23 - When does execution start? (e.g. On a write command some or all of the data is moved across the interface. Has the write command started execution or not?)
2. Page 11 - Pending Task should be defined in the glossary
3. Page 15-16 - Several of the entries in the symbols list do not have any spaces between the second column and the third column.
4. Page 39 Section 3.7 1st~~st~~ paragraph after figure 17 near the end of the paragraph there is an undefined cross-reference.

5. Page 42 - The heading 4.1 and the two lines following it are duplicated on the top of page 43.
6. Page 44 - 1st~~st~~ paragraph after table 3 - The second to the last sentence should have a cross-reference to clause 6.6.1.

7.*Page 44 - 1st~~st~~ paragraph after table 3 - This is the start of the ACA description problems. There should be a statement in this paragraph that says that if the ACA bit is zero the SCSI-2 rules for handing exception conditions shall be used by the Target. No further definition is required and in fact the definitions in clause 4.6.1 only confuse anyone who has implemented CA in SCSI-2.

8. Page 46 - section 4.2 - command terminated paragraph - There is a undefined cross-reference at the end of the first sentence.
9. Page 46 and other places throughout the document - Statuses and messages have been changed from ‘Queue’ to ‘Task Set’. Was this change agreed to by the committee? If so OK if not it should be voted on.
10. Page 50 list under entry b - There should be an ‘or’ between RESET, TARGET RESET.
11. Page 53 section 4.6.1.1 last paragraph - The first sentence should be changed from ‘the mode byte of the’ to ‘the control byte of the’.
12. *Page 53 section 4.6.1.1 last paragraph - The statement ‘shall be unconditionally cleared upon receiving the next command from the faulted initiator’ is not the behavior described in SCSI-2 for this condition. (See comment number 7 for the solution.)
13. *Page 54 section 4.6.1.1 first paragraph last sentence - What are the conditions for ‘acceptance’? Where in the task set is the command placed; is the task treated as a Head of Queue, Simple, or Ordered task?? (See comment number 7 for the solution)
14. *Page 54 section 4.6.1.1 2nd~~nd~~ paragraph first sentence states ‘The completion of the new task with’. I do not know what is meant by ‘new task’ in that sentence. I assume it is an attempt to reword the 2nd~~nd~~ paragraph in section 2.1.1 of the SCSI-3 Queuing Model but the message seems to have been lost.
15. *Page 54 section 4.6.1.1 3rd~~rd~~ paragraph first sentence should be changed to ‘faulting command, then the auto contingent allegiance condition shall not be cleared and a new task shall be entered into the..’
16. *Page 54 section 4.6.1.2 3rd~~rd~~ paragraph last sentence: I have not idea what this sentence means.
17. *Section 4.6.1: After careful study of this section there seems to be several concepts defined in the SCSI-3 Queueing Model that are not here. The Missing concepts are list below:

2.1.2 Response to Auto Contingent Allegiance Condition

If a Task becomes a current task because of a previous

request for information that information shall be suspended until the ACA is cleared.

2.1.3 Auto Contingent Allegiance Processing

All SCSI operations are permitted while processing an ACA

Task.

2.1.4 Clear Auto Contingent Allegiance Task Management Function

The target shall clear the Auto Contingent Allegiance and

complete the current Task on acceptance of this task management function.

If the target accepts a Clear Auto Contingent Allegiance

Task Management Function and no Auto Contingent Allegiance Condition is in effect for that initiator on that task set, then the target shall complete the current Task.

If a Clear Auto Contingent Allegiance Task Management

Function occurs when an ACA Task is pending then the ACA Task shall be aborted and the auto contingent allegiance shall be cleared.

Attempted shall be used.h then an ASC of Overlapped Commands

18.* Page 54 section 4.6.2 last paragraph 2nd~~nd~~ sentence should be changed to '...tasks for the initiator that caused the overlapped commands in the task set...'.

19. section 4.6.2 - The list of things that can occur to free up tags is not listed. The list out of the SCSI-3 Queueing Model follows:

    2.2 Duplicate Tag Handling

    When issuing a tagged task the initiator shall not

    reuse the tag to create a new task until:

    - A service response of Command Complete is received with a status other than INTERMEDIATE or INTERMEDIATE-CONDITION MET.
    - A service response of Service Delivery or Target Failure is received. In this case, system implementations shall guarantee that the task associated with that command has been terminated.
    - A power on condition occurs.
    - A Target Reset Task Management request occurs.
    - An Abort Task Management request occurs.
    - An Abort Task Set Management request occurs.
    - A Clear Task Set Management request occurs.
    - A unit attention of TASKS CLEARED BY ANOTHER INITIATOR is reported.
    - A unit attention of POWER ON, RESET or TARGET RESET is reported.

20.* Page 58 section 4.6.5 - In this section it must me made clear that the clearing of the unit attention condition does not automatically clear the auto contingent allegiance condition if the ACA bit is set to one.

21. Page 62 section 5.3 last paragraph - The last sentence should be changed to ' ...subject to the task set management...' and the wrong clause os referenced; it should reference clause 6.
22. *Page 67-74 sections 6.2 - 6.4.4.1 - I attempted to understand this section to see if it matched the SCSI-3 Queueing Model but I could not. The problems I had were:
    - Cross-references in almost every sentence.
    - Figure 23 is incomprehensible to me (Bring back the bubbles).
    - The words previous and prior drive me crazy what's wrong with simple words like before and after.
    - Sections 6.3.1 to 6.3.8.1 are frustrating trying to understand.

      To net it out: I cannot determine if SAM complies with the SCSI-3 Queueing Model. The only thing I can go by is section 6.5 which contains the task management examples. Everything in the examples looks correct except for one minor editorial change (see below). But those are only examples and I cannot assume the other sections correctly define the actions within the examples.

23. Page 76 figure 29 - The lower left task set brackets for the enabled tasks should be extended to include the Head of Queue (Task 7).

Quantum comment:

We do not believe that the compliance issues raised in X3T10/94-081R0 have been adequately addressed.

X3T10/94-081r0 is included here for reference:

Background

SAM is a standard for standards, and yet it is very unclear to me what is mandatory and what is optional within SAM, and whether the referenced item is one within a standard claiming SAM compliance or an implementation of that standard.

Proposal

I therefore suggest the following modifications to SAM (rev 12A):

(new item) Page 12: 2.1.28 implemented: The referenced item from a standard is realized in the actualization of the standard. For standards complying with SAM, an implemented item is an item in SAM realized within that standard. For devices complying with a standard (e.g. products), an implemented item is an item in that standard which is realized within that device.

(new item) Page 13: 2.1.34 item: A feature defined within a standard. The fundamental element for claiming compliance to a standard.

Page 13: 2.1.41 mandatory: The referenced item is required to claim compliance with the specified standard.

page 13: 2.1.44 optional: The referenced item need not be implemented to claim compliance with a specified standard.

(new item) page 13: 2.1.45 option set: A set of referenced items which individually are considered to be optional. If all of the items are implemented, then the option set is implemented. If any item is not implemented, then the option set is not implemented.

A new section should also be created (I suggest section 2.4) as follows:

Grounds for Compliance

SAM consists of a set of items, each of which is clearly noted as being mandatory (either explicitly or implicitly by the use of the word “shall”) or optional (either explicitly or implicitly by the use of the word “may”). Optional items may be further grouped into sets of options (“option sets”).

All standards claiming compliance with SAM shall implement within the standard all mandatory SAM items. A standard may implement optional SAM items.

All standards claiming compliance with SAM shall in turn consist or a set of items, each of which is clearly noted as being mandatory or optional. Optional items may be grouped together as optional sets. Any implementation claiming compliance with that standard shall implement all mandatory items and may implement any optional items.

Any standard claiming compliance with SAM must provide a facility to allow the determination of whether an implementation of that standard implements specific optional items identified in SAM. This determination shall typically be done during a configuration period.