digital™ StorageWorks

Summary of Proposed Responses to SAM Revision 12 Review Comments

January 19, 1994 X3T10/94-028R0

From: Charles Monia
SAM Technical Editor

To: Members of X3T10

Subject: Summary of Recommended Responses to Review Comments on SAM, Revision 12

The enclosed document contains my recommended responses to the review comments received from the following members of X3T10:

Jeff Williams (Hewlett-Packard),
George Penokie (IBM),
John Lohmeyer (NCR),
Steve Heil (Panasonic),
Gene Milligan (Seagate),
Bob Snively (Sun Microsystems),
Jeff Stai (Western Digital)

All comments are included verbatim. Proposed responses are preceded by a right arrow ">". A comment with no response indicates an item to be incorporated as written.

From: Charles Monia
SAM Technical Editor

To: Members of X3T10

Subject: Proposed responses to Hewlett-Packard Comments on SAM, Revision 12

Begin proposed response

#001 (E) Page 2, Ed Note

Remove the editors note.

#002 (E) Page 3, 3.4 and 4.7.3.2

Fix formatting of the lines (make the .....17 and .....36 on the same lines as the 3.4 and 4.7.3.2).

#003 (E) Page 9, Section 1, Bullet 10

SSA-SCSI should be SSP. Remove the question mark.

#004 (E) Page 10, Section 1, Para 6

Add a definition to the SMC section as was done in all of the previous sections (e.g. SPC, SBC, SSC).

#005 (E) Page 10, Section 1, Para 12

Again, not SSA-SCSI, but SSP. Also, add that SSP runs over the SSA-PH protocol.

#006 (E) Page 10, Section 1, Para 14

SPI is the SCSI-3 Parallel Interface, not Interconnect.

#007 (I) Page 10, Section 2

This section must be filled in. I have a problem forwarding any document with a TBD section. This should be a minor item to correct.

#008 (I) Page 11, Section 3.1, General

Use the agreed upon definitions from 92-141R8 wherever applicable.

> The agreed upon definitions will be used as noted above.

#010 (E) Page 20, Section 3.5, Para 2

Shouldn't the "[input-1,input-2,...]" be "{input}" indicating zero or more input parameters. This is consistent with the notation in the previous section.

>
> The suggested notation ("{...}") is used to defined a composite
> object. I acknowlege the potential for confusion and will consider
> modifying the notation accordingly. I am reluctant, however, to extend
> object annotation to procedure calls.
>

#012 (T) Page 23, Section 4.2.1, Para 2 and 5

Paragraph 2 states that there is one application client for each task. Paragraph 5 states that an application client controls one or more tasks. Which one is correct? I thought that the first case is correct - that there is one and only one task per application client.

>
> The intent of the sentence was to point out that certain requests,
> such as CLEAR TASK SET, may effect several tasks. The text will be
> modified to eliminate the misleading wording.
>

013 (T) Page 27, Section 4.3.3

I do not think that we need an LLP Service. First, it is not a good model in that a ULP should never control an LLP at another node. Second, the premise that data transfer is not a ULP request

is not true. Data has to go somewhere in the control of the ULP. The ULP does not have to give up control to the LLP for this transfer to take place. I do not see the need for this section.

>
> The specified service type will be deleted. The Data Delivery
> services will be modiified to use a peer-to-peer model.
>

#014 (T) Page 31, Section 4.6, Para 1

You cannot require that the order be preserved in all cases for a given pair of devices. For example, if I run in fibre channel over a fabric and send two command frames, the order is only guaranteed if I request in-order transmission over the fabric. I may not want to do this for performance reasons. I think that you need to say that order may be "imposable" in the cases where ordered tasks are sent or some other ordering is required, but you cannot require it in all cases.

>
> I believe that, although some physical transports, such as Fibre
> Channel, may reorder data in transit, the ordering specified by the
> sender is restored before the transaction is presented to the consumer.
>
> In any event, ordering is implicit in SCSI-2 today. Therefore
> placing a new ordering burden on the application client (e.g., the
> device driver) may lead to implementations that break existing
> code that would otherwise be portable.
>
> I suggest this issue be left open for discussion
> at the next working group.
>

#015 (T) Page 35, Section 4.7.3.1, Para 1

The task manager controls the execution and "responds to" task management functions. It does not control execution "in response to" task management functions. The latter implies that a task management function is always required to control a task, which is not the case.

>
> The intent of the paragraph was to specify the role of
> the task manager, namely: to service task management functions,
> whose purpose is to control task execution in some way.
>
> The text will be modified to make the above point explicitely.
>

#016 (E) Page 38, Section 5, Para 2

The "Object Identifier" is not included in the description in section 3.5. Also, this appears to be a duplication of the description in section 3.5. Remove this section if it is a duplicate.

>
> The redundant material will be deleted.
>

#017 (T) Page 38, Section 6, General

You also need an optional input and output arguments for protocol specific information. For example, FCP returns more information in status than just sense data and a status byte. It also returns residual byte counts. There is no place for this in the ExecuteCommand primitive.

>
> The intent of the specification is to preserve software portability
> through protocol-independance. In my opinion, protocol-specific
> parameters are, therefore, inappropriate for SAM interfaces.
>

#018 (T) Page 39, Section 6, Para 4

The last sentence states that the buffer is undefined for a status such as CHECK CONDITION. This is not true in all cases. I can return data from a read and then run into an error and return CHECK CONDITION. The data is valid in the buffer and may even contain the block in error if the TB is set. I may also have recovered all data via ECC and returned the data along with a CHECK CONDITION indicating that the data was recovered with ECC. In either case the data is valid.

>
> The section should be modified to state that, in the absence of
> supplementary information supplied by the device, the
> application client shall consider the buffer contents to be undefined.
>

#019 (E) Page 39, Section 6, Para 5

Remove the "???". Replace it with 6.5.8.

#020 (E) Page 39, Section 6, Para 6

Typo. No close paren on the paragraph.

#021 (T) Page 39, Section 6, Para 11

Again, the buffers may be valid even though a CHECK CONDITION status was returned. Remove this restriction.

>
> See response to item 018.
>

#022 (T) Page 40, Section 6.1.1, Para 1

The wording from SCSI-2 to this document changed so that I am no longer required to check any reserved fields! In SCSI-2, it states that, "A target that receives [reserved fields] that are non zero...". In SAM it states, "A logical unit that detects a reserved field that is not zero..." These are totally different! According to SAM, I don't have to do anything unless I check and detect the non-zero fields. This is less restrictive than SCSI-2.

>
> The section will be corrected as pointed out above.
>

#023 (T) Page 40, Section 6.2, Para 3

Are there specific commands which will alter the medium when errors are present in the command descriptor block or parameter list? This was not allowed in SCSI-2. Why is it allowed here?

>
> The section will be corrected as noted above.
>

025 (E) Page 42, Section 6.2.1, Para 3

Add references to tables 1, 2, and 4 like was done for table 3.

>
> During the SAM review, the working group agreed that CDB formats
> should be moved to SPC. All the CDB figures will be deleted from
> SAM and replaced by a single generic diagram.
>
>

#026 (E) Page 42, Section 6.2.1

Why aren't logical block address, transfer length, parameter list length, and allocation length described?

>
> See reply to item 25 above.
>

#027 (T) Page 43, Section 6.2.2, Para 2

The flag bit is optional even if the link bit is supported. There is a new requirement in this paragraph which states that the flag bit is required if the link bit is supported. This was not required in SCSI-2.

>
> The specification will be corrected as noted above.
>

#028 (E) Page 43, Section 6.2.2, Para 5

Typo. Extra period in the last sentence.

#029 (T) Page 43, Section 6.3, Para 2

The list should also include ABORT TASK and ABORT TASK SET.

#030 (T) Page 43, Section 6.3

SCSI-2 allowed the status to be sent prior to the CDB. This clause was removed in SAM. It should be included in appropriate wording for the protocols (e.g. don't use terms like status phase).

>
> It seems to me that the point at which status may be returned should
> be defined within the applicable protocol standard. I don't believe it's
> appropriate for inclusion in SAM.
>

#031 (E) Page 45, Section 6.3, Para 2

TASK SET FULL states that it is required if tagged tasks are

supported. Tagged tasks support is required, therefore remove
the statement about it being optional (the first sentence).

>
> As I understand it, device support for tagged tasks is still optional.
> If it is not optional, then devices such as tape drives will have to be
> modified for SCSI-3 compliance. ie. As a minimum, even if such devices
> are limited to one task per logical unit, they will have to understand
> queue tag messages and the like.
>
> I recommend discussing the matter at the next working group.
>

#032 (E) Page 45, Section 6.3, Para 2

Also add into the TASK SET FULL status that the task is discarded.

#033 (T) Page 48, Section 6.5.2

There is a paragraph from the 92-141R8 document missing from this clause. This paragraph described the reaction to an ACA condition being created within an ACA by an ACA tagged task. Include this paragraph.

>
> The requirements in the specified paragraph will be included.
>

#033 (T) Page 48, Section 6.5.2

There is a paragraph from the 92-141R8 document missing from this clause. This paragraph described the reaction to an ACA condition being created within an ACA by an ACA tagged task. Include this paragraph.

>
> The requirement called out above is included in clause
> 6.5.2.2, paragraph 3.
>

#034 (E) Page 49, Section 6.5.2.1, Para 3

This is only the case if the ACA bit was set in the control byte. If the ACA bit was clear, this should be an illegal queue type. I assume that the ACA bit means that the error handling is just like in the SCSI-2 standard which means that the next command clears the CA. By the way, auto sense should be treated as though it was a Request Sense after the CA went into effect thus clearing the CA.

>
> Autosense will be defined as noted above.
>
> The following agreement was reached after the working group:
>
> If the ACA bit was clear in the faulting command, then
> any new task will clear the ACA condition, regardless
> of task attribute.
>

#035 (1) Page 49, Section 6.5.2.2, Last set of bullets

If the ACA bit was zero, then the error handling is strictly SCSI-2 and should not be defined further. The ACA bit was meant to provide a backwards compatibility. Let's not redefine what that means. I think all that needs to be said is that auto sense is treated as a command following the CA condition. This should be stated in the section on auto sense.

No additional information should be given for an ACA bit cleared.

>
> See response to comment 034.
>

#036 (E) Page 50, Section 6.5.3

This entire section should be replaced with the section from 92-141R8. There are a number of additional reasons why the tags may be reused. In addition, the use of the two ASCs is explained in 92-141R8.

>
> The requirements from the specified section will be added to
> SAM.
>

#037 (E) Page 50, Section 6.5.4, Para 5

General comment. Whenever sense data is discussed, mention it in general terms of the sense key, additional sense code, etc. Do not say, "In response to a Request Sense command...." since auto sense makes this an invalid statement.

>
> Agreed.
>

#038 (E) Page 50, Section 6.5.4, Para 6

Typo. "Logical unti" should be "Logical unit".

#039 (E) Page 53, Section 6.5.8

State that the auto sense acts as an implied Request Sense command in order to eliminate the confusion on handling the ACA bit cleared in the control byte.

>
> Agreed.
>

#040 (E) Page 53, Section 6.6, Para 1

Why is hard reset condition "[to be specified]"?

>
> Hard reset will be defined in a protocol-independant
> manner.
>

#041 (E) Page 53, Section 6.6, Para 9

Bullet (h) should be "Any other event that requires the attention of the initiator". Leaving it as "Any other event" tells me to generate a Unit Attention when anything happens!

>
> Agreed.
>

#042 (E) Page 54, Section 6.6, Para 1

What if the Inquiry has been sent after the ACA was created? In SCSI-2 it was cleared. Is that what happens here also? I assume that the Inquiry is treated like any other command and that an ACA ACTIVE status is sent if it is not tagged with an ACA Tag Type.

>
> Inquiry is treated like any other command.
>

#043 (I) Page 54, Section 7.1, Para 2

The definition of suspended information should be the one from 92-141R8.

>
> Agreed.
>

#044 (E) Page 54, Section 7.1, Para 3

The definition of current task should be the one from 92-141R8.

>
> Following the working group, it was agreed to specify in SAM that the
> definition of a current task is protocol specific.
>

#045 (E) Page 54, Section 7.1, Para 4

The definition of pending task should be the one from 92-141R8.

>
> Agreed.

#046 (T) Page 55, Section 7.2, Para 11 (Dormant)

What is execution mean? This has no meaning to me. Can I bring in write data for a write command? Is this executing?

>
> See following comment.
>

#047 (T) General Comment.

All of the text associated with Head of Queue, Ordered, and Simple has been removed from SAM. This should not have happened! The text was succinct. It described exactly what was to complete and when. We have now gone back into this nebulous "execution" of tasks model that failed in SCSI-2. I cannot vote to forward anything that allows this model to exist.

>
> References to "execution" will be removed wherever there is a
> possibility of ambiquity. In all other respects, the queuing
> model presented in SAM must, of course, be in full technical
> agreement with the version passed by the committee.
>
> Comments are, of course, appropriate in specific cases where
> SAM is not in agreement.
>

#048 (T) Page 57, Figure 22 and Page 58, Para 2

SIMPLE TASKS do not wait for previous SIMPLE TASKS to complete before executing when restrictive reordering is applied. They are only restricted if an overlap occurs on the data set being operated upon. The restriction also only applies to write CDBs.

>
> The specification will be corrected as described above.

#049 (E) Page 57, Figure 22

A note is needed stating that entry to the figure is restricted by an ACA condition. For example, if an ACA is in effect, the HEAD OF QUEUE tasks are not allowed to be submitted to the task set. If this note is missing, then the figure implies that HEAD OF QUEUE may coexist with ACA tagged tasks during an ACA.

>
> The text will be modified to explain that the state diagram only
> applies to tasks entered into the task set while no ACA condition is in
> effect.
>

#050 (T) Page 57, Figure 22

What can be done in each of these states? Can I bring in write data in the Dormant state? Can I bring in write data in the Held state?

>
> The allowed behavior for dormant tasks will be added as
> required by 92-141. The behavior for other tasks states is
> given in the accompanying text on pages 58 and 59. I believe
> this text is in full agreement with the requirements specified
> in 92-148.
>

#051 (T) Page 61, Section 7.4.1, Figure 24

Depending upon your definition of executing, I do not agree with the boundaries applied. I can bring in data for an ordered write command when simples are being operated upon.

>
> The allowed behavior for dormant tasks will be added to
> the specification. According to 92-141R8:
>
> "When a task is an Ordered Task any information the target
> has or accepts for the Ordered Task shall be suspended and
> ---------
> the Ordered Task shall not complete until all Simple and
> Ordered Tasks that were accepted into the task set before
> the Ordered Task have completed."

>
> It appears that the behavior described in the first sentence
> quoted above applies to all tasks in the dormant state. That
> sentence seems to imply that the device server may solicit data
> for dormant tasks.
>

#052 (T) Page 67

Restrictive reordering only applies to commands with overlapping LBA ranges. It does not apply to all simples unless they meet the overlapped criteria.

>
> The specification will be corrected as specified above.
>

#053 (E) Page 70, Section 8.3, Para 5

Add statement that the target shall guarantee that no further responses from the aborted tasks are sent to the initiator.

>
> Agreed.
>

#054 (E) Page 70, Section 8.4

What happens on a CLEAR ACA when an ACA does not exist?

>
> The CLEAR ACA is a NOP in that case. According to 92-141R8:
>
> "If the target accepts a Clear Auto Contingent Allegiance
> Task Management Function and no Auto Contingent Allegiance
> Condition is in effect for the initiator on that task set,
> then the target shall complete the current task."
>

#055 (T) Page 74, Section 9.2

Why is this an LLP service? The LLP has no idea where the data is located. It is a confirmed service between two ULPs.

>
> The LLP service will be changed to a peer-to-peer service
> between ULP peers.
>

#056 (T) Page 75, Section 9.2, Para 1

The assumption that the client buffer is contiguous is not necessarily true. How I put my data in my buffer is completely outside the scope of this standard. It is true that the data buffer is a logically contiguous space, but not physically.

The buffer also does not have to be large enough to physically hold the data. I may page out data to some other space or consume the data as it is read into a buffer.

>
> The specification will be modified to require that the
> buffer be logically contiguous.
>

From: Charles Monia
SAM Technical Editor

To: Members of X3T10

Subject: Proposed Response to IBM Comments on SAM, Revision 12.

The following contains the proposed responses to IBM's comments on SAM revision 12. Each comments is included verbatim, followed, when appropirate, by the proposed response, which is denoted by a right arrow ">".

A comments with no response will be incorporated as specified.

Begin Proposed response
==========

| * There is a fundamental change between the Queueing Model and the way SAM defines Queueing. It is that the Queueing Model assumes the target is the 'owner/manager' of the task sets. This allows the possibility of task set boundaries other than on a per Logical Unit with no change to rules of the queuing model other that the boundary definition.
|
| SAM defines the 'owner/manager' of task sets to be the device server. There is one device server per Logical Unit. And the device server has no knowledge of or control over other logical units within a target. Because of this there is no way a device server could manage a task set which crosses logical units.
|
| There is nothing 'wrong' with the way SAM has defined the task set control because the 'one task set per logical unit' was what the committee voted to support. I am only pointing out that with SAM it will require a much greater effort to change to a different or support for multiple task set boundaries. I other words we are burning bridges.

>
> The definition of "task set" will be decoupled from the
> definition of logical unit while retaining the concept of
> one task set per logical unit.
>

2 * Through out SAM there are references to the confirmation of services and the response to services. This is confusing: Are services confirmed or responded to?

>
> Two kinds of services are defined:
>
> 1. Distributed services performed by servers within the the target and
>
> 2. Protocol services provided by the lower layer protocol.
>
> Distributed services consist of a request which elicits a server response.
>
> Protocol services consist of a protocol request which may elicit a
> confirmation from the lower layer.
>
> The editor will review the specification and correct any ambiquities
> in terminology.
>

3 Page 10 Section 2: TBD Must be removed

4 Page 11 Section 3.1.11: The sentence 'A command has than has..' should be changed to 'A command that has...'

5 Page 12 Section 3.1.21: The sentence '...for the next in a series...' is not clear and should be changed to '...for the next task in a series...'

>
> Agreed.
>

6 * Page 12 Section 3.1.26: This definition implies that application clients are part of initiators. I do not agree. application clients are the originators of commands and initiators do not originate commands.
Page 23 section 4.2.1: Again refers to the application client residing in initiator devices.

>
> The working group agreed that the definition in the specification was
> acceptable.
>

7 Page 12 Section 3.1.28: What is an integrated path? I can find no definition of it.

>
> The working group agreed to the following modified definition:
>
> Interconnect subsystem: One or more physical interconnects which
> appear as a single path for the transfer of information between
> between SCSI devices in a domain.
>

8 Page 12 Section 3.1.30: The sentence '...commands executed by a single task..' is not correct. Tasks do not execute things. I suggest changing 'executed' to 'contained within' or 'bounded by'.

>
> Agreed.
>

9 Page 13 Section 3.1.41: The sentence '...devices in a domain.' should be changed to '...devices within a domain.'

>
> The working group agreed to accept the wording in the specification.
?

10 Page 14 Section 3.1.61: Why is this definition even in SAM? None of the other protocols are defined. If this definition stays then all the other protocols need to be defined. If it stays the the definition should be 'A protocol defined by the SIP standard.' nothing more.

>
> The referenced definition will be deleted.
>

11 Page 14 Section 3.1.63: This is a GS definition. Remove everything after '...device delivery subsystem.'

>
> Agreed.
>

12 Page 15 Section 3.1.73: What is a 'constituent system'? What is a 'related system'? What has this definition have to do with SCSI?

>
> The word "constituent" will be deleted.
>

13 Page 15 Section 3.1.74: The sentence '...the data was addressed.' should be '...the data is associated.'

>
> Agreed.
>

14 Page 15 sections 3.1.77 and 3.1.79: The first word of both sentences should be 'An'.

15* Page 15 section 3.1.84: The logical unit does not define the boundaries of the task set. The sentence '...grouped by the logical unit so...' should be '...grouped so...'.

>
> Agreed.
>

16 Page 15 section 3.1.85: Task slots do not 'represent' tasks. I think think of task slots 'holding' tasks.

>
> The working group agreed to the following revised definition:
>
> "task slot: Resources within the logical unit that may be used to
> contain a task."
>

17 Page 19 figure 1: The is a line missing on the 'section' block.

18* Page 33 section 4.7.1: Pending is being used but it is not clear what is meant. What does not mean to have a 'pending task management function'? Depending on the definition of 'pending' it may or may not be possible to have pending task management functions.

>
> A task managment function is pending from the time it is issued by
> the application client to the time when a target response is received.
>
>

19 There seems to be an inconsistency in the usage of 'task' and 'command'. If I read the definitions of each it would not be possible for the application client to deal with tasks. But for linked commands I know this not the case because there are many commands which make up a single task. So if I ignore the definitions then the description of 'Application Client' on page 33 is OK. But now I go to page 34 and read the description of 'Logical Unit' and it talks about commands not tasks. How is anyone supposed to understand this.

>
> As explained during the working group, the application client issues
> commands and task management functions. In response to a command,
> the device server may create a task that provides the context within
> which a command executes.
>
> The working group felt that the definitions and usage seemed
> consistent.
>

20 Page 34 section 4.7.2: The sentence in 'Logical Unit' '...request are directed.' should be '...requests are directed.'

21 Page 35 section 4.7.3: The sentence in 'Device Server' should be '...SCSI commands and manages the task set.'

>
> Agreed.
>

22 Page 35 section 4.7.3.1: The last sentence in the second para. the 'should' should be 'shall'. And here is the undefined pending task management again.

>
> The working group agreed to retain the wording of "should".
>

23 Page 36 section 4.7.3.2: In the 'Tagged task ' section: Why not make life a little less complicated and change 'initiator-specified component (tag)' to 'tag' or at least 'initiator-specified tag'.

>
> Agreed.

>

24 Page 39 section 6: In the 'autosense data' section there is an undefined cross reference.

25 Page 39 section 6: In the 'Austosense Return Flag' section there is a missing ')'.

26 Page 39 section 6: In the 'Status' section: When is the status undefined?

>
> As stated in the argument description, status is undefined whenever
> the command fails to complete with a service response of command complete.
>

27 Pages 40,41, 42, 43, and 44: All the Tables on these pages are messed up.

28* Page 44 table 8:The committee requested that this table be changed to remove the possibility that anyone would interpret status codes as being bit sensitive but coding the statuses as hex values.

>
> The change will be made as stated above.
>

29* Page 45 section 6.3: In the 'ACA Active' section the two places where 'command' is stated should be 'task'.

>
> I believe the references to "issuing a command" in the first
> sentence and item b are correct. In item C, the reference to command
> should be changed to task.
>

30* Page 46 section 6.4.1: Bullet number 2 talks about a task being created when it is received by the device server. Again; how can the application client know about some that does not exist?

>
> As I read it, there is nothing in item 2 that implies such knowlege on the
> part of the initiator.
>

31* Page 49 section 6.5.2.1: Paragraph 2: The sentence '...was created by the initiator...' should be changed to '...was created for the initiator...'. An ACA is not created by an

initiator, it is created for an initiator.

>
> Agreed.
>

32* Page 49-50 section 6.5.22: This is section is a disaster. There appears to be an attempt to redefine how SCSI-2 CA should work and it is not right. I will argue that it is a bad idea to try to define a SCSI-2 function is SCSI-3. We should just say 'see SCSI-2'. Look at the 'Queueing Model document to see how this is handled'

>
> The working group agreed that behavior must be described explicitly
> without refererence to SCSI-2.
>
> Subsequently, the editor agreed to modify 6.5.2.2 to delete item B
> and revise item C to unconditionally clear the ACA when any new task
> is created, regardless of task atttribute.
>

33* Page 50 section 6.5.3: This section is not reflect the committees desires for overlapped commands. See section 2.2 of the queueing model for the latest information.

>
> This will be corrected to agree with 92-141R8.
>

34 Page 50 section 6.5.4: Bullet b - The sentence ' ...the logical unti...' should be '...the logical unit...'.

35* Page 52 section 6.5.7: Is Asynchronous event reporting per target?

>
> Asynchronous event reporting is per logical unit.
>

36 Page 53 section 6.6: In the first paragraph there is a '[to be specified]': What is that all about?

>
> A protocol-independant definition of hard reset will be added.
>

37 Page 53 section 6.6: Bullet h: What is 'Any other event'? That statement could imply that an unit attention could legally occur on any event. (eg command complete, check

condition, etc.)

>
> This itemization will be changed to:
>
> "Any other event requiring the attention of the initiator."
>

38* Page 54 section 7.1: The 'Ordered Blocked Boundary' section looks to be an attempt to combine the 'Ordered Blocking' and 'Task Set Boundaries' concepts. These two concepts are important to understand and should be separately defined.

>
> The definition of Ordered Blocking Boundary in SAM will be
> corrected.
>

39 Page 55 section 7.2: In the 'task abort' section and elsewhere the message names have changed from Queue to Task. I do not think this is a good idea and yes in know task is the 'correct' term but if this change is made it will be forever confusing.

>
> Since the SCSI-3 queuing model replaces "queue" with
> "task set", the working group agreed to the names for task
> management functions that are currently in the specification.
>

40 Page 56 section 7.2 last sentence: The sentences read '... to the commands it completes...' What is the 'it' this section of the sentence is referring to?

>
> "It" refers to the antecedent "task". The sentence will be reworded
> to replace the pronoun.
>

41 Page 58 table 58: In the Dormant row under the 'All prev. HOQ and previous ORDERED tasks ended' column 'tasls' should be 'tasks'.

42 Pages 61,63,65, and 67: The Black background with white lettering is in many cases almost unreadable.

43* Page 73 section 9.1: The send SCSI command looks allot like the execute command service response but some of the parameters are in different locations. Are there supposed to be two different services that are almost the same? If so why?

>
> The paramaters in the Send SCSI Command service will be changed to agree
> with the parameters of the Execute Command remote procedure call.
>

End proposed response

N

The following are proposed responses to the NCR commnents on revision 12 of SAM.

1. The figure just after "1 Scope" should be numbered.

2. The reference to SCSI-2 in the second to last paragraph of the Scope Clause should be "X3.131-199x.

3. The Scope should include a statement that additional SCSI-3 documents may be added to the architecture without SAM necessarily being updated to reflect the addition.

4. All TBD's must be removed (that is, completed) before a document goes to public review.

5. In 3.1.1, either restate this definition to eliminate the term "ACA attribute" or define "ACA attribute" in the glossary.

>
> The term will be defined in the glossary.
>

6. In 3.5, Notation for Procedures and Functions, the usage of () and [] differ from their usage in 3.4.1, Object Notation. Readers could become confused. I suggest aligning these two notations by always using [] for optional items and always using () for item lists.

>
> While I agree there's potential for confusion, I don't see
> an easy way around this problem since both notations are
> based on well-established conventions.
>

From: US3RMC::"John.Lohmeyer@FtCollinsCO.NCR.COM" "John Lohmeyer" 4-JAN-1994 03:55:29.14
To: Charles Monia <starch::monia>
CC:
Subj: Re: Proposed responses to your review comments on SAM Rev 12.

"Charles Monia, SHR3-2/W4, 237-6757, monia@starch.enet.dec.com 11-Dec-1993 1536 wrote:"
>
> The following are my proposed responses to your comments on revision 12
> of SAM. Following your review, I plan to post this on the SCSI reflector.

I hope you didn't wait for my response to post your proposed responses. In any case, at l o n g last, here is my reaction to your proposed responses.

> begin proposed response
> ==========
> The following are proposed responses to the NCR commnents on revision 12
> of SAM.
>
> Comments are included verbatim, followed by responses, when applicable.
> The responses are preceded by a right arrow ">". Comments with no
> response will be incorporated as specified.
>

Since you basically agreed with me on 1-5, I have only a positive reaction to your responses.

> 6. In 3.5, Notation for Procedures and Functions, the usage of
> () and [] differ from their usage in 3.4.1, Object
> Notation. Readers could become confused. I suggest
> aligning these two notations by always using [] for
> optional items and always using () for item lists.
>
> >
> > While I agree there's potential for confusion, I don't see
> > an easy way around this problem since both notations are
> > based on well-established conventions.
> >

I would agree that both notations have been used, but I don't think both are commonly used in the SAME document. That is the point of my comment. Why does SAM have to use BOTH conventions? Are we being compatible with the notation that exists in two external documents? If so, we should reference these documents and state why our notations are consistent with the external documents rather than being self-consistent.

>>
>> In my opinion, the way around the problem is to ensure that the context in which each
>> is used is clear to the reader so that it is possible to distinguish which notation is in effect.
>> In that light, I'd apprciate specific instances where such confusion has been encountered.
>>

From: US3RMC::STARCH::MONIA "Charles Monia, SHR3-2/W4, 237-6757, monia@starch.enet.dec.com 03-Jan-1994 1508" 3-JAN-1994 16:47:55.09
To: scsi_reflector@Pa.dec.com
CC: starch::monia
Subj: Proposed response to Panasonic Comments on Revision 12 of SAM

Begin proposed response
F==========

R1. Though the SAM document has been through several major revisions there is still significant work needed for the document valuable to the industry. This is particularly the case with Annexes A and C which represent a significant amount of committee effort but are not consistent with the remainder of the document. Concepts like queuing and terminology such as "execute", "task", "response", "confirmation" are not consistent. The document will confuse readers in its present state.

> SAM must, of course, be in full technical agreement with the queuing
> model passed by the commitee. Any discrepancies or inconsistencies betweem
> the body of the document and the queuing model will be corrected. In that
> regard, please cite specific instances where the draft is either
> unclear or at variance with the queuing model.
>
> It is my understanding that once such corrections are made annex C will be
> deleted. The commitees' intent regarding the alternate task set descriptions
> in annex A, however, is not clear to me. I propose that the issue of
> whether or not to retain annex A be resolved in the working group."

R2. The document requires the use of "Per Logical Unit Task Set Boundaries" but discusses and provides for other implementations. This is very confusing. My experience is that these options in a standard will become requirements in the marketplace and therefore should be better documented in the standard. If the intent is to provide extensibility through these options it should be clearly stated.

>
> I believe the discussion of other alternatives in annex A was in accordance
> with the commitees' wishes. Please see the previous response.
>

R3. I am confused by the requirement in clause 4.6 that all transactions be received in the order they were sent. My understanding was that some of the SCSI-3 transports do not maintain order (i.e. P 1394, Fibre Channel and possibly SSA).

>
> I believe that, although some physical transports, such as Fibre
> Channel, may reorder data in transit, the ordering specified by the
> sender is restored before the transaction is presented to the consumer.
>
> In any event, ordering is implicit in SCSI-2 today. Therefore
> placing a new ordering burden on the application client (e.g., the
> device driver) may lead to implementations that break existing
> code that would otherwise be portable.
>
> I suggest this issue be left open for discussion
> at the next working group.
>

Charles Monia
From: Charles Monia
SAM Technical Editor

TO: Members of X3T10

Subject Proposed responses to Seagate Comments
on SAM Revisiion 12

Begin proposed response

==========

===

From: US3RMC::"Gene_Milligan@notes.seagate.com" "Gene Milligan" 3-JAN-1994 15:17:55.51
To: starch::monia
CC:
Subj: SAM Comments

1/3/94
John Lohmeyer
Chairman X3T9.2

Cc: Charles Monia Technical Editor

Subject: Comments on Working Draft SCSI-3 Architectural Model (No Document Number) Revision 12

1) A minor nit, on the front page, the copyright notice should not mention ANSI X3.131, since this dpANS has no resemblance to that document and therefore that copyright could have no relevance.

2) The foreword spelling should be corrected and it should serve a more useful purpose than inviting questions. The technical interpretation should not be deleted but it should be preceded with some introductory information regarding SCSI-3 and SAM. Global Engineering documents should not be in the body of the standard - not even the foreword.

3) I think CAM should be added to Figure Anonymous in the scope. In listing (10) the description should be updated and the (?) deleted. In listing (12) it should be corrected to "Common" rather than "Command".

>
> It has been suggested to me that this matter should be discussed with

> the folks in the CAM Working Group. I recommend raising this issue
> during the upcoming working group.
>

4) Delete "under X3T9.2 jurisdiction".

5) Delete "target" from SPC.

6) SBP is not the IEEE SerialBus. It is the Serial Bus Protocol for IEEE 1394.

7) For GPP, the interconnects are not packetized. GPP paketizes them. If it is not made a technical report, delete "packetized".

>
> This may be splitting hairs, but I believe the interconnects that GPP runs on
> do their own packetization based on the physical characteristics of the medium
> (e.g., ethernet). Although GPP may pass information in packets of it's
> own, the interface to the physical media may split such packets into
> chunks that are compatible with the media.
>
> In any event, I feel the current figure and descriptive text
> is correct as it stands. The issue is, of course, open to debate at
> the next working group.
>

8) In all instances of 1394 delete "P".

9) In the next paragraph change SCSI to SCSI-3 since SCSI-1 and SCSI-2 do not comply with SAM.

10) SCSI-2 is not "-1992". It may be "-1993". Check with yourself to see if it is 1993 or 1994.

11) The TBD of section 2 needs to be addressed.

12) In 3.1 the ACA attribute needs to be defined or a forward reference given.

>
> A generic reference to the definition in section 3.2 (acronyms and abbreviations) will be
> added.
>

13) In 3.1.9 "aborting" does not define "aborted". What does "execute" mean?

>
> A definition for the term "abort" will be added.
>
> "Execute" has the standard dictionary meaning. The following is quoted
> from the 1973 edition of "The American Heritage Dictionary":
>
> "Execute: ....To perform; do....To carry out .... in accordance with a
> prescribed design".
>

14) In 3.1.20 what is "starting execution"?

>
> See previous reply regarding the definition of "execute".
>

15) In 3.1.24 Replace "defined" with "listed" and delete "may be specified by the system integrator or". It would be better if "implementation-specific" were replaced with "vendor-specific" as used in SCSI-2.

>
> I propose the following wording:
>
> "Implementation Specific: A requirement or feature described in this
> standard, whose implementation or behavior may be specified by the
> systems designer or is vendor-unique."
>

16) The movement of the SCSI-2 initiator to encompass more Host functions is troublesome.

>
> I don't see the problem here. Can you eleborate?
>
> I believe the usage in SAM is consistent with the SCSI-2 definition
> given below (SCSI-2 Spec, rev 10h):
>
> "3.1.17: initiator: An SCSI device that requests an I/O process to be
> performed by another SCSI device (the target)"
>
> As I read it, the reference to "device" in the above defintion seems to
> embrace not just the interface to the physical media, but the
> controlling intelligence as well.
>

Do application clients have a close relationship to applications?

>
> The application client represents the request-specific "thread" of
> execution within the initiator that is independant of the actual interconnect.
> In an implementation, that thread would include the application-specific intelligence,
> the device driver and any lower level common code that manages I/O requests without
> knowlege of the physical interconnect characteristics.
>

Does an application taking advantage of multiple initiators for resiliency have multiple application clients or application clients using multiple initiators?

> Yes - One way is multiple application clients using multiple physical busses
> and interfaces.
>
> Another is a single application client using a service delivery subsystem that has
> redundant elements for availability. Of course, to the client, such redundancy would
> be invisible, except for the avaliability improvement.
>
> One thing to bear in mind is that what's being specified here is simply a reference
> model. That is, it's a vehicle or 'metaphor' that I use to describe behavior.
> A target, for instance, has to behave as though it has an entity called a logical unit.
> Even though that entity may not correspond to some physically identifiable component.
>

17) In 3.1.28 how does the single integrated path relate to full duplex SSA, multi pathing GPP, and maximum surprise FC fabrics? An analogous question applies to 3.1.41.

>
> First, I've been asked and have agreed to remove the word 'integrated'.
>
> Anyhow, the notion of a single path simply says that to the talker and listener, the
> channel behaves as though it were a single 'wire', even though the information packets
> may be routed over one or more different physical paths. Whether or not that wire is
> full or half duplex is of no concern to the SCSI entities conversing over the channel.
>
> Keep in mind that SCSI conversations are inherently half-duplex. That is,
> a given SCSI command consists of nothing more than a series of requests
> and responses.
>

18) In 3.1.32 what is the significance of "externally"?

>
> The defintion should be chaged to read as follows:
>
> "logical unit: A target-resident entity which implements a device model and executes
> SCSI commands sent by an application client.
>

19) How does the LUN definition of 3.1.33 apply to a system based upon SPI?

>
> SPI is only capable of responding to a subset of the address space. According to
> SAM section 3.4.2, such subset behavior is allowable.
>

20) The definition of "optional" should go on to state the standard verbiage
about if it is implemented it has to comply with the definitions of the
standard.

21) In 3.1.74 what does "concealed from the task" mean? Isn't there a more
direct definition.

>
> The definition in annex C says "information that is not available to the task". I have
> agreed to use that definition. Although, I felt that the notion of information being
> concealed was stronger. To me, 'not available' implies that the task may be aware of
> the information but not allowed to use it - sort of like a hungry animal in a cage
> that can see food but can't get at it. In that case, the awareness may effect behavior.
>
> Concealment, on the other hand, imples that the task is totally unaware that the
> information exists. Hence, there can be no effect on behavior.
>

22) I have the impression that targets receive and LUNs execute (whatever that
means). In 3.1.75 it is difficult to follow whether the target or the LUN is
being referred to.
>
> I'll reword this.
>
23) In 3.1.84 I think "queuing" should be replaced with "task management".

>
> Agreed.

>

24) Regarding 3.1.87 I thought the third party command could also involve an initiator acting upon the behalf of another initiator.

>
> The temporary initiator, at some point, must act like a logical unit so it
> can receive the commands which it will subsequently execute as an initiator.
>

25) In 3.2 has SCSI-3 written off SCSI-1?

>
> No.
>

26) In 3.3.1 why is neither the USA or the ISO numeric convention used?

>
> The conventions used here were copied from the SCSI-2 spec.
>

27) In 3.4 I don't think conceptually needs to be in quotations.

28) In 3.4.2 replace "must be" with "shall be". "implementation" should be replaced with "protocol".

Is there actually a case in this standard of a
maximum where the address or identifier can not be less? Perhaps "this standard" should be replaced with "the protocol standard".

>
> There are no cases (so far) where SAM requires a protocol to support the
> maximum. However, I'd like to reserve the privelege of doing so at a later
> time. ie. I recommend retaining the existing wording.
>

29) Delete the small file names in each of the figures or move them to a corner and add to the conventions what they are. Another alternative would be to make them white on white.

30) In 4.1 It is difficult to understand how the common set of requirements in SAM can apply to interconnects which were all developed prior to SAM.

>
> I see no conflict here.
>
> SAM's job is to specify what is required for an I/O system to be in compliance
> with SCSI-3. Given a 'menu' of existing interconnects, SAM provides the criteria
> for selecting those that can be used to build an SCSI-3 conformant system.
>
>

31) In 4.2 should there be a definition for "caller".

>
> In my opinion, this needs no special definition.
>

32) In 4.2.1 hopefully task management functions provide more use than aborting or terminating. For example I think they also allow performance optimization through command overlapping (in the English sense - not the SCSI overlapped sense).

>
> As I read that section, the reference to aborting a task was in the context
> of an example. I don't see that exmaple as implying that such functions are
> limited to terminating tasks.
>

33) In the last line of 4.2 I doubt that 4.4 defines each system component.

>
> I don't see the problem here. System components, in the context of SAM, are
> only those defined in the structural model.
>

34) Figure 9 seems to imply that the I/O System Model is a loop. Perhaps there should be a note to indicate that other than a loop is accommodated.
>
> Figure 9 was intended to show how a physical implementation can be abstracted to
> produce a generic domain topology.
>
> Perhaps some added words are necessary for those literal-minded folk.
>

35) In 4.6 the last line could benefit from different terminology than "order". It implies that chaos is also allowed.

>
> How about "ordering".
>
> The sentence would then read:
>
> "The architecture does not require that ordering be preserved between transactions from
> different source devices"
>
36) The page layout should be adjusted to allow Figure 13 and some of the other figures to be larger for readability.

37) The Application Client box in Figure 15 is incomplete.

38) Is it true that an initiator can have as low as 0 Application Clients? (Is the system turned on?)

>
> Yes, it may be turned on but idle. Application clients only exist while work is being
> performed.
>

39) In 4.7.2 why the disparity between Logical Unit and Target. Why isn't the definition Logical Unit Number or Target? (Why Target Identifier?)

>
> That section describes the elements that comprise the 'target' object. I don't
> see the disparity you speak of.
>
>

40) Regarding the note in 4.7.3.1, why isn't this true even if other requests are not pending?

>
> The issue is the problem of possible dependencies between a series of pending task
> management requests. For instance, if a commmand fails, the initiator can abort
> other tasks that may have depended on the results produced by the faulting
> command. In the case of pending task management functions, hgwever, the initiator has
> no such recourse.
>

41) Why doesn't 4.7.3.2 have an upper bound of one on untagged tasks per initiators?

>
> It's not necessary to specify this constraint explicitly.
>
> It is impossible to have more than one untagged task per "I_T_L' without
> having a duplicate task identifier.
>

42) Another nit, the comma in 6 should be outside the bracket enclosing "Autosense data".

43) The ??? needs to be replaced with a section number.

44) "(see clause 6.5.8" needs a closing bracket.

45) Under Autosense Return Flag - Status replace "shall be undefined" with "is not valid".

>
> I believe 'undefined' is more appropriate. To me, 'not valid' implies that
> something is illegal. Clearly, an undefined quantity is legal under
> these circumstances.
>

46) In the last line replace "The actual events" with "The protocol events".

47) In the title for 6.1 delete "Implementation".

48) In the first and third instance change SCSI-3 to SCSI. Also change "future revision to" to "future version of".

49) In the first paragraph of 6.2 delete "one of".

50) Change "All SCSI protocol specifications shall accept" to "All SCSI protocol standards shall define".

51) In 6.2.1 why is Group 5 the only one with a table reference?

>
> All format descriptions will be moved to the SCSI command standards.
>

52) Should the Group 6 and 7 vendor specific commands now be limited to 16 or less bytes?

>
> I believe that issue should be discussed in the working group. Assuming
> the commitee concurs with removing the CDB format defintions from SAM,
> I assume this would then become an issue to be addressed by the command
> standards.
>

53) In 6.2.2 does the citation of of protocol standards include CAM and SCSI-2?

>
> In my opinion, CAM is not a protocol standard.
>

54) What does item (d) of 6.3 mean?

>
> A non-recoverable error detected by the target, possibly due to
> a fault in the service delivery subsystem, that cannot be
> reported to the initiator.
>

55) I remain concerned that there may be a conflict with the installed base which had a presumption of bit significance and the use of the BUSY bit in the Task Set Full Status code.

>
> Please feel free to raise the issue in the working group.
>

56) If the committee again rules out bit significance of the installed base, the bit significant language of Intermediate_Condition Met should be changed.

>
> Agreed.
>

57) In 6.5.1 the second paragraph is too definite on what can be changed. Change "The parameters that can be be changed by" to "Among the parameters than might be capable of being changed by".

>
> I have been asked, and have agreed, to delete this section in the
> next draft. It is my understanding that the corresponding material
> will be moved to the SPC document.
>

58) The last sentence of the second paragraph seems to denigrate the recommended procedures of SCSI-2 to "vendor specific cases". Why is this?

>
> Assuming this section is moved to SPC, as noted above, this issue
> should be discussed in the context of that document.
>

59) In the third paragraph change "should" to "shall".

>
> See response to item 58.
>

60) The next two paragraphs need to be challenged on the basis of "execution".

>
> See response to item 58.
>
> In my opinion, there is no ambiquity in the use of the term here.
> I believe it's reasonable to assume that the command is in the
> process of execution at some point and has completed execution
> when a status is returned.
>

61) Change the last phrase to "optionally descriptive".

>
> See response to item 58.
>

62) In the last sentence of 6.5.2 delete "construed to mean".

63) In 6.5.2.2 I think the ACA should be cleared by a power off condition rather than power on.

>
> I'm not sure why the distinction is important. I don't think the
> initiator can tell the difference anyhow.
>
> In my opinion, it's difficult to imagine that a target would have the time to execute
> useful work during power down. I think it's conceptually more useful
> to imagine clearing state on power-up.
>

64) It appears to me that the next to last paragraph should be included after item (a) (the second (a)).

65) Why should a protocol standard not require overlapped commands (in the SCSI sense) to be detected? (6.5.3).

>
> The rationale is that, in some protocol implementations, checking for
> command overlap has too much overhead due to the large tag address
> space.
>
> I suggest discussing this matter during the working group.
>

66) In 6.5.4 should the " e.g." statement be "one peripheral device" or "one LUN"?

>
> Logical Unit.
>

67) In the last sentence of 6.5.8 replace "client must issue" with "client shall issue".

>
> I disagree. The word "must" is appropriate since the client is not
> required to issue the request sense. ie. The client can do something
> else and still comply with the standard.
>

68) In 6.6 the "(to be specified)" needs to be replaced.

69) I recognize that the wording of item (f) is the same as SCSI-2. However, is the intention that the Unit Attention condition be set even if there was no change to the parameters?

>
> I recommend discussing this matter in the working group.
>

70) The wording of (h) has changed. Shouldn't "that requires the attention of the initiator" still be included? Otherwise the interfaces will do nothing but service events they do not need to.

>

> The wording will be fixed.
>

71) In the fifth paragraph item (1) change "on the logical unit" to "for the logical unit".

72) In item (2) delete "may" or it collapses to item (1).

73) In the last paragraph delete "on the logical unit".

74) In 7.1 there is approximately an infinite amount of information concealed from the task but most of it is not suspended information. A better definition is needed for suspended information.

>
> The definition in annex C, section 0.1 is:
>
> "suspended information: Information stored within the logical unit
> which is not available to any pending tasks. Task management functions shall
> not be suspended."
>
> I propose that this definition be used the basis for replacing
> the corresponding text in 7.1.
>

75) The concept of an Ordered Blocking Boundary should not be enclosure but should be a separation boundary. Comment also applies to Figures in section 7.4.

>
> Agree.
>

76) In the second line of 7.2 change "is a description these" to "is a description of these".

77) A page break, as the document stands now, would be beneficial ahead of "Events:"

78) A reference should be given to the location of the QErr bit definition.

79) I consider issuing "Command Complete" part of "command execution". I suggest Enabled be "The task may continue processing and issue a service response of Command Complete."

>
> Agreed.
>

80) Assuming a task can not be simultaneously Held and Blocked, I suggest replacing "The state of a once" with "The new state of a".

>
> Agreed,
>

81) I don't consider TERMINATE TASK executed until the task is ended and has no state. What if the ACA was for the task of the TERMINATE TASK? My informed poll did not uncover any implementors of TERMINATE TASK.

>
> The definition needs to distinguish between when
> issuance of the terminate task request is complete and when the
> terminate task operation has been performed.
>

82) Why is Dormant singled out for the extra discussion? An exception to the description is the time from created to completed which is observable. Therefore the "shall" requirement can not be met.

>
> I don't understand your point here. Interactions between tasks are
> observable and hence can be specified. The text was trying to make
> the point that the results produced by a successfully completed task
> should be independant of whether or not that task was previously
> in the dormant state.
>

83) In 7.3 the description of what Figure 23 and Table 10 are don't seem to match the nomenclature of the figure and the table.

>
> I'm not sure what problem you're citing. The reference to the
> figure 23 and table 10 in section 7.3 (second sentence, pp 57)
> says:
>
> "Transitions for ACA tasks or tasks entered into the task set while
> an ACA is in effect are shown in Figure 23 and table 10."
>
> I don't see the problem with that text.

>
84) It can not be deduced from 7.2 that an isolated (the only command this week) SIMPLE task starts in the Dormant state. I thought it started Enabled.

>
> SIMPLE tasks start in the Dormant state "by definition". If there
> are no previous ordered or head of queue tasks, then a SIMPLE task
> instantaneously enters the Enabled state.
>
> I'll modify the text accordingly if that will clarify matters.
>
85) A small nit, change "As shown Figure 22," to "As shown in Figure 22,".

86) In the line ahead of Table 9 delete "be".

87) In the third from last paragraph, Termination Pending, it is not clear what caused the second ACA.

>
> The second ACA occurs when a terminated task completes with
> TASK TERMINATED status.
>

88) An exception to the next to last paragraph is if the state were already Done.

89) In 7.4 it does not appear that the Legend box is needed. If It were deleted the figures could be bigger and more readable. Less shading would also help readability.

>
> Agreed.
>

90) In the dreaded IMPLEMENTORS NOTES of section 8, change "all initiators on all task sets" to "all initiators in all task sets".

91) In 8.1 change item (2) to "begins processing the function."

92) In the last paragraph of 8.4 change "When" to "After". (Fewer carriage returns above that paragraph might take care of the orphan problem for "Description:" in 8.7.)

93) The third from last paragraph, counting the note, of 8.7, suffers from

confusion as to which function or task is being referred to. Consequently I concluded that if the target could not stop a task, the task would be completed. But if it didn't support the function the task could not complete in a normal manner. I think the paragraph can only be understood if the reader already knows how it works and can fault tolerantly process the terminology.

94) Another nit, "(Autosense Buffer Pointer,)" should have the comma after the parenthesis.

95) I had awarded the "a/an" prize to the SBP editors but perhaps you should share it. Because it is so distasteful I can not tell you what to do about the "n" ahead of "SCSI" in the second paragraph of Annex A.

>
> Of course, the typo shoud be fixed. I assume we will, for lack of
> anything better to do, continue to debate whether the acronym SCSI
> should be preceded by the article "a" or "an".
>

96) Annex C should be deleted and any needed information, if not covered by SAM already, included in SAM.

>
> Agreed.
>

G.E. Milligan

--
Gene Milligan (Gene_Milligan@notes.seagate.com)

T
Begin response to SUN review comments

E 001 Page 9 and 10, Section 1.0

The reference document identifications should be completed, including at least the official title and project draft number.

E 002 Page 11, Section 3.1.11

The sentence should be corrected in the area: "command has than has ended".

E 003 Page 16, Section 3.1.9

The word "word" should be re-examined. If all goes well, a search throughout the document will reveal that the word is either never used or can always be replaced with something like "4-byte value", avoiding ambiguity.

E 004 Page 16, Section 3.3

From context, it appears that lower case is used both for words having their normal English language meaning and for words other than SCSI commands, statuses, and acronyms. The last sentence should either be deleted or the distinction between lower case and upper case significance made clear.

E 005 Page 23, Section 4.2.1 and Page 33, Section 4.7.1

Page 23 indicates in the next to the last paragraph that an application client controls one or more tasks.
Page 33 indicates in the definition of application client that there is one application client for each task or pending task management function.

It is my impression that there is a one-to-one relationship between application clients and tasks and that each new task is created by a new instance of an application client. That would make page 33 correct and page 23 incorrect. This should

be clarified in both locations and corrected in one.

>
> The text will be modified to make the one-to-one relationship clear.
>
> References to the control of multiple tasks will be deleted.
>

E 006 Page 24, Figure 4

Figure 4 uses a layering model which does not refer to the layering terms defined immediately below it. I believe that the LLP and ULP terms should be properly assigned to the figure. That either requires relabeling or may require that the central layer be broken into a ULP layer and an LLP layer.

>
> Agreed.
>

T 007 Page 31, Section 4.6

The last paragraph of section 4.6 indicates that the service delivery transactions are received in the order in which they are sent for a given pair of source and destination devices. Fibre Channel and some other channels allow the proper operation of SCSI with out of order delivery of command information. This restriction should be modified to allow the out of order delivery of commands if operating system or channel conventions can guarantee the proper behavior of the scsi targets. As an example, ordering of groups of commands can be enforced by the host adapter function or by management of individual commands with respect to the acknowledgment processing of commands requiring ordering.

>
> I had assumed that, while data may arrive out of order, the receiver
> would restore ordering before presenting the data to the consumer.
>
> I am concerned that relaxing the ordering requirement in the manner
> suggested will lead to implementations that break existing host code which
> depends on the implicit ordering provided by SIP/SPI.
>
> I would like to reopen this issue at the next working group.

>

E 008 Page 33, Section 4.7.1

Shouldn't the Initiator equation be:

Initiator = 0{Application Client} + 1{Initiator Identifier}1

If not, a considerable amount of additional information is required to indicate what the rules are for the execution of tasks across multiple independent ports.

>
> Multiple identifiers should be considered as nothing more than
> an alias for the same physical entity.
>
> I would like input from the working group on this issue. According
> to past feedback, multiple identifiers for the same entity were
> considered acceptable.
>

E 009 Page 34, Section 4.7.2

Shouldn't the Target equation be:

Target = 1{Logical Unit} + 1{Target Identifier}1

If not, a considerable amount of additional information is required to indicate what the rulse are for execution of tasks to a LUN having multiple target ports within a single task.

>
> See reply to comment 008.
>

T 010 Page 35, Section 4.7.3

LUN 0 should always exist, at least to the point of being able to respond to TIO, INQUIRY, and REQUEST SENSE. The LUN need not be installed.

>
> Agreed.
>

T 011 Page 36, Section 4.7.3.2

The Task Set should have only one Untagged Task per initiator according to previous definitions. The definitions in this document allow more than one, since an untagged task can also have an identifier. If it has an identifier, isn't it tagged?

>
> The identifier for an untagged task does not included the tag parameter.
> Thus, by definition, an initiator cannot request the creation of more
> than one untagged task per logical unit without causing a duplicate
> task I/D.
>

T 012 Page 37, Section 4.7.3.2

The last sentence implies that the only mechanism known at the service layer and at the communication level might be the Task Identifier. In fact, the Task Identifier known at the service layer may have an arbitrary mapping to the Task Identifier information actually passed across the protocol interface. As an example, CAM knows a service request by the pointer to the CAM Control Block, but a SIP implementation uses the initiator identifier, target identifier, LUN, and Tag to perform the identification of the same IO Process. This should be clarified.

> The implication was intentional.
>
> Commands such as copy require the application client to have
> explicit knowlege of the target identifier and logical unit number
> used by the protocol.
>

T 013 Page 39, Section 6

The Data-In Buffer Pointer is an input accompanying the Execute Command, not an output. It identifies the buffer that will be used when the Data In Delivery Service is executed.

>
> The convention for denoting inputs and outputs will be modified to
> distinguish between a container to receive an output and

> the output itself.

T 014 Page 39. Section 6 and page 74, Section 9.1

The Autosense Data, as an output arguement, is not defined elsewhere. In particular, it is missing from the definition of section 9.1, the Confirmation returned to the Application Client. The referenced clause should be defined.

>
> The autosense return flag indicates whether or not sense data was
> returned to the autosense buffer. Input from the working group is
> is needed to determine the conditions under which autosense data
> is to be returned. ie. Can autosense data be returned for any command
> or only those which complete with a status of CHECK CONDITION or
> COMMAND TERMINATED?
>
> If the latter, then this flag is unneccesary.
>

T 015 Section 6 and Section 9

The Command Model and the Service Model appear to be partially uncorrelated and redundant. I would suggest that Section 6.1 be replaced by Sections 9.1 and 9.2, with appropriate complementary information retained from section 6.1. That would become a new section 6.

I would then suggest that section 9.3 be moved to become the new section 7.

The sections 6.1 - 6.6 would become the new section 8 and section 7 would be renumbered to section 9.

This would restructure the document to present the model information in conjunction with the service model that implements it.

A summary of the chapter titles would be:

5: Procedural model for commands and task management functions

6: SCSI Command Model and Services

6.1: Data Transfer Services

7: SCSI Task Management Model and Services

8: SCSI Command and Protocol

9: Task Set Management

10: Task Management Functions

If this cannot be done, a great deal of editorial work must be done to show the relationship between the Command Model and the Service Model and to correlate them.

>
> The document will be restructured to resolve the inconsistencies
> mentioned.
>

E 017 Page 39, Section 6

The Autosense Return Flag should be labeled the Autosense Returned Flag.

>
> Agreed.
>

E 018 Page 40, Tables 1, 2, and 3.

Using unjustified, uncentered, courier type would make this work better. Alternatively, it should be presented using the table functionality of the word processor.

T 019 Page 43, Section 6.3

Status is also not presented after Abort Task and Abort Task Set.

T 020 Page 47, Section 6.4.2

In the execution of linked commands, step 5, the accessing of

the second command, is normally created by actions of the target. This is especially obvious in SIP, where the target manages the command phase of the second command directly. It is also true for SBP, where the target fetches new commands from initiator memory and for the FCP, where initiative is passed to the initiator to enable the output of the command information.

>
> The document will be modified to state that the manner in which the
> next command is solicited is protocol specific.
>

T 021 Page 48, Section 6.5.1

The Programmable Operating Definition is not a characteristic of SAM or of any protocol. It can actually be used to influence only those things that are visible at the command set level, and so should only be described in the command set. In particular (ref paragraph 2), the SCSI compliance level, SCSI specification level, and other parameters should be explicitly banned from changing, since these parameters reflect details like the use or non-use of an Identify message, the use of extended Identify messages, and the width of data transfers, items which may extend into the requesting and the responding services.

Among the other parameters which should not be changeable are included:

Vendor Identification: Changing this parameter allows fraudulent replacement of disk drives with non-conforming disk drives.

Vendor Serial Number: Changing this parameter allows fraudulent warranty repairs to be sought.

The technical solution to this problem is to remove the section from SAM and to place it in the appropriate section of the SPC document. The CHANGE DEFINITION command should then be described in such a manner as to prohibit modifications to anything other than command level behavior and to prohibit modifications that create the opportunity for fraud.

>
> This section will be moved to SPC in its entirety.
>

T 022 Page 48, Section 6.5.1

The text, when taken over to SPC, should contain a precise indication of the state of the machine necessary to accept and execute a CHANGE DEFINITION command. That state should be that no tasks for any initiator are present in the task set of the logical unit. This can be accomplished by reserving and quiescing the logical unit through normal command set operations. The CHANGE DEFINITION command should be rejected if any tasks are in the task set for the destination LUN.

The text, when taken over to SPC, should contain a precise indication of when the new definition becomes active. It should become active at the time that the service response indicating successful completion of the CHANGE DEFINITION command is transmitted from the LUN.

>
> These matters should be addressed to the SPC technical editor.
>

E 023 Page 49, Section 6.5.2.2

The definition of "hard reset" should be documented.

>
> Agreed.
>

E 024 Page 50, Section 6.5.3

According to Annex C and SCSI-2, the proper response for overlapped commands should be a sense key of "ILLEGAL REQUEST", not "ABORTED COMMAND".

>
> The correction will be made as specified above.
>

E 025 Page 50, Section 6.5.4

Item B, typo: "unti" should be "until"

T 026 Page 52, Section 6.5.7

It should be clearly indicated that AEN is an optional behavior, both by the target and by the initiator. While a protocol is required to architect the capability, no device should be required to implement the capability.

>
> Agreed.
>

T 027 Page 52, Section 6.5.7

In the last paragraph of the page, the text indicates that AEN should be reported only once per occurrence of the causing event. In fact, for errors that are generic and may influence the operation of any attached initiators, the AEN should be presented to all attached initiators. The text should be modified to clearly indicate that AEN should be offered only once to the initiator related to the command causing the failure, but is allowed to be offered to every initiator if it is unclear to the target which initiator will be affected by the failure.

>
> I believe this requires discussion by the working group at large.
>

T 028 Page 52, Section 6.5.7

AEN is defined in SCSI-2 as sharply limited in its capabilities. It is always executed as an initiator-to-target operation, where the target is identifiable through the INQUIRY command as a Processor Type device. The information is sent as a SEND command to LUN 0 of the device.

In particular, AEN SEND cannot be generated to a target that identifies itself as other than a Processor Type device because the command overlaps the decode of a WRITE(6) in other command sets.

Text should be installed in section 6.5.7 describing this characteristic of AEN.

>
> Since the current specification makes it clear that the delivery
> mechanism for asynchronous event reporting is protocol-specific, I

> believe no further information is required.
>

E 029 Page 53, Section 6.6

The events, items a-h, should include additionally those unitemized events in the first paragraph, including TARGET RESET, hard reset, and power-on reset.

>
> Agreed.
>

T 030 Page 54, Section 7.1

The definition of "current task" and therefore "pending task" varies somewhat from the definition contained in Annex C. In Annex C, the current task is defined as a task that is moving data to or from the physical transport, meaning in the case of disk drives and tapes, that data is being moved to or from the magnetic storage medium. In fact, the term "current task" is not used in section 7 and should probably be dropped. Of the three terms used, only one has meaning in the clarified task management structure, and that is Ordered Blocking Boundary. That term should probably be moved into section 7.2 and section 7.1 should be collapsed out of existence.

>
> I believe the definition of current task in annex C refers to a task
> that is actively transferring data over the physical interconnect.
>
> Anyhow, in discussing this matter during the last plenary, it was decided
> that SAM should require each protocol standard to define the concept of a
> "current task".
>

E 031 Page 54, Section 7.2

First paragraph, undetermined information missing in second sentence.

E 032 Page 55, Section 7.2

"All Previous tasks complete" event needs to be clarified. In particular, it cannot occur for simultanously enabled tasks and is only true at ordered blocking boundaries.

>
> "Previous tasks" are all tasks that are older than the task in
> question. In view of that definition, it's not apparent to me where
> the problem lies.
>

E 033 Page 55, Section 7.2

The definition of the QErr bit must be obtained from SPC and installed here, since it is not clear what meaning it will have without it's being defined.

>
> Agreed.
>

T 034 Page 55, Section 7.2

Task abort should include other causes of task termination, including hard reset and power on reset.

>
> Agreed.
>

T 035 Page 55, Section 7.2

There does not appear to be a state for tasks that are on-going behind the scenes, including immediate commands and tasks that have been aborted as far as the SCSI interface is concerned, but are continuing on to completion to bring the logical unit to a known state. These may not actually be treated as tasks, but rather as peer applications started by tasks. In any case, mention of their allowability should be made in section 7.2.

>
> I believe the scope of SAM must be limited to describing
> required task behavior that is visible to the application client.

> In that regard, once a task abort completes, the task ceases
> to exist as far as the initiator is concerned.
>
> While the peer processes you describe may be relevant to the
> internals of specific target implementations, I don't think
> SAM should have anything to say about them.
>

T 036 Page 57, Section 7.3

Restricted reordering appears to be defined incompatibly with SCSI-2. In SCSI-2, the restricted reordering only requires that the final value of all data observable on the medium shall have exactly the same value as it would have if the commands had been executed in the same received sequence without using tagged queueing (which in SCSI-2, meant "received synchronously".) SAM appears to require the execution in precisely the specified order in the case of restricted reordering. This is an overly strict requirement and should be modified to the SCSI-2 definition. As an example, that means that READ operations having the SIMPLE task attribute performed under restricted ordering may be executed in any order.

This must also be corrected in paragraph 2 of page 58, same section.

>
> This material is incorrect and will be deleted from the specification.
>

E 037 Page 57, Section 7.3

The state diagram applies to each single task, but does not express the relationship among tasks. This should be clarified in the title and explanations.

>
> Agreed.
>

E 038 Page 58, Section 7.3

The last paragraph before Table 9 has a typographical error in the last sentence. The word "be" should be deleted.

T 039 Page 58, Section 7.3

The restrictions on head of queue task enabling are more correct than those in section 7 of Annex C, which are probably unenforceable.

T 040 Page 59, Section 7.3

The second sentence of the last paragraph may be overly general. The CLEAR ACA should only abort tasks if the QErr bit is set to one.

>
> This should be discussed at the next working group. The description
> in the specification reflects inputs from others in the
> working group.
>

T 041 Page 60, Section 7.3

During an ACA condition, only properly offered ACA tasks are enabled. All other tasks are rejected. This is not made clear in the text or table.

Alternatively, it may be required for a task to enter the enabled state for it to be rejected. If this interpretation is correct, the state diagram should be modified to contain it.

>
> According to the model, the task is created by the target, then tested
> to see if it can be accepted into the task set.
>
> The specification will be amended to clarify the above.
>

T 042 Page 61, Section 7.4

The examples pertain only to those LUN's having a queue management capability. It is also possible that typical tape drives, sequential data delivery devices (voice recorders, etc.), mechanical devices (like media changers), and certain other devices may not allow tagged queueing at all. In this case, the SAM should absolutely not force the devices to perform queue management,

but rather should allow the synchronous delivery of the tasks under control of the initiator's application client.

This is allowed almost everywhere, but an example should be included in section 7 to demonstrate that it is allowed.

In particular, such devices have a slightly different behavior with respect to overfilling their very restricted task set. A second task will be treated as an overlapped command, rather than a Task Set Full condition. This must be made clear in sections 6.3 and 6.5.3.

>
> Agreed.
>

T 043 Page 63/64, Section 7.4.2

It is not clear that tasks 4,5, and 6 should be enabled before task 7 (a head of queue task) is completed. I would not expect any new tasks to be enabled until all head of queue tasks are done. See step 3 in the sequence.

>
> According to the queuing model (92-141R8):
>
> "Any Task that was accepted into the task set before a
> Head Of Queue Task may complete before the Head Of
> Queue Task completes."
>

T 044 Page 67, Section 7.4.4, step 1

The text should read:

1) Since restricted ordering is in effect, simple task 1 can be enabled. If the effects of enabling simple task 2 would cause a violation of the restricted reordering algorithm, it must remain dormant until simple task 1 is completed.

It is likely that, if both simple task 1 and simple task 2 are read operations on a disk drive, they can both be enabled and

still meet the restricted ordering requirements.

>
> The inclusion of Restricted Reordering into the queuing model is
> in error. All references in SAM will be deleted.
>

T 045 Page 68, Section 8

SIP only provides an indication that the task management function has been delivered, not that it has been completed. It is possible that the Function Complete service response is stepping beyond the original definitions from SCSI and should be softened to an accepted delivery indication.

>
> The bus free following successful delivery of the last message byte is the
> SIP/SPI completion indication.
>

T 046 Page 70, sections 8.2 - 8.7
(Modification required to change vote to affirmative).

It is clear from the document that each protocol shall be required to provide a mechanism to perform each of the task management functions. In addition to this, it must be made absolutely clear which of the task management functions are optional for a SCSI device to implement and which ones are required. Some task management functions may only be required if certain other optional capabilities are allowed.

Terminate Task is an example of a function that is always optional.

Clear Task Set is an example of a function that is only required if the task set elects the definition:

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

Clear Auto Contingent Allegiance is an example of a function that is only required if the ACA bit is allowed to be set to one.

These are probably best placed in a new paragraph under each Task Management Function entitled "Service Response"

A typical case would be for Terminate Task:

Service Response:

Function Complete:	Indicates Terminate Task Function was accepted and will be attempted by Device Server
Function Rejected:	Indicates Terminate Task Function not implemented by Device Server
Failure:	Indicates Terminate Task Function could not be delivered to Device Server

A contrasting case would be for Clear Task Set:

Service Response:

Function Complete:	Indicates Clear Task Set Function was accepted and will be attempted by Device Server.
Function Rejected:	This response is only allowed for Device Servers that reject all Tagged Tasks.
Failure:	Indicates Clear Task Set Function could not be delivered to Device Server.

Table of desired optionality:

Task Management Function	Opt/Rqrd	Conditions
Abort Task	Rqrd	
Abort Task Set	Opt	Rqrd if Tagged Tasks supported
Clear ACA	Opt	Rqrd if ACA bit = 1 supported
Clear Task Set	Opt	Rqrd if Tagged Tasks supported
Target Reset	Rqrd	
Terminate Task	Opt	

>
> This issue needs to be discussed at the next working group.
> Others are of the impression that support for tagged queuing
> is mandatory for all devices.
>

T 047 Page 70, section 8.2, 8.3

Should the Abort Task or the Abort Task Set function including a task with the ACA attribute also abort the ACA condition?

At present, the answer is it should not be.

>
> That is correct. The following excerpt from 92-141R8
> specifies the conditions under which an ACA shall be
> cleared
>
> "The Auto Contingent Allegiance Condition shall be cleared
> after:
>
> -a power on condition,
> -a Clear Auto Contingent Allegiance Task Management request
> issued to the logical unit by the initiator receiving the
> Check Condition or Command Terminated status error.
> -a Target Reset Task Management request
>
> The Auto Contingent Allegiance shall not be cleared for any
> reason other than those listed above."
>

T 048 Page 70, section 8.2

What should the Device Server do if an Abort Task for a task with a certain task identifier overlaps or arrives shortly before the task with that task identifier? There is some uncertainty about whether the new task is the one to be aborted or is indeed a new task that is not to be aborted. This is theoretically possible with certain drivers and out-of-order delivery fabrics.

I would suggest that any task with the specified identifier be aborted if it arrives at the device server any time from before the task management function arrives until the service response from the task management function is transmitted by the device server. Any identical task after that time should be considered a new task. A recommendation that task identifiers be maintained unique for some time period after completion of the task would make this requirement more robust.

>

> In my opinion, the behavior specified in SAM should be based on a
> sequential delivery model. An SDS implementation
> may, of course, do what it likes provided the behavior visible to
> application clients, device servers and task managers is sequential.
>
> See the reply to item 007.
>

T 049 Page 70, section 8.4

When does the Clear ACA take effect? I propose that it becomes effective immediately upon the arrival of the Clear ACA request.

>
> Agreed.
>

T 050 Page 71, section 8.5

The phrase "All data for all terminated tasks shall be cleared." is ambiguous. I assume that this includes both state information and the contents and identification of data buffers. In particular, if a WRITE operation was active when a Clear Task Set was taking place, the data that had been transmitted to the disk, even if incomplete, would be the data returned by a subsequent READ. The buffered data, even if incomplete, would not be returned by the READ.

>
> Since tt is effectively impossible for an initiator to know the
> exact point at which a task was aborted. the present wording is
> is adequate, in my opinion.
>

T 051 Page 71, Section 8.6

The definition of hard reset is not present in the SAM document.

>
> A definition for hard reset will be added.
>

T 052 Page 71, Section 8.6

In SIP, the target address, rather than the Logical Unit, is the destination of a TARGET RESET function. It is probably reasonable to change the definition to LU Identifier, but this is definitely a change in the structure of SCSI. I would further propose that for those devices having a hierarchical Logical Unit structure as described in the RAID model, that the TARGET RESET be defined to reset not only the specified Logical Unit, but all Logical Units depending from that Logical Unit in the hierarchy.

>
> The model will be changed to define a single task manager per target.
> The Target Reset function will then require only the target
> identifier.
>

T 053 Page 72, Section 8.7

The definition of corrupting the medium should be clarified. In particular, none of the Abort functions are expected to stop so quickly that incorrect CRC is written on the recording medium. All of the functions, including Terminate Task, leave the medium in an ambiguous state in that the operation is incomplete. Properly implemented, Terminate Task could provide the necessary information to make it possible for the host to determine the nature of the medium corruption, but cannot avoid the corruption.

>
> The reference to "media corruption" will be replaced with:
>
> "The TERMINATE TASK" function is invoked by the application client to
> terminate the specified task and return its state of completion."
>

054 Page 72, Section 8.7

In the third paragraph, item 1, there is a discussion of the residue of an allocation length or parameter list under the conditions of a Terminate Task. Terminate task has some meaning with respect to data transfer tasks, where there is a long operation which will be interrupted. For tasks

that are instantaneously executed or which require the complete parameter list for execution (MODE SELECT and similar commands), the meaning of Terminate Task should be clarified. I believe that the command should either not begin execution or should complete execution regardless of how many bytes have been transferred.

If the task completes successfully, it should not indicate TASK TERMINATED status, but should indicate function complete (GOOD status).

>
> Trying to define the termination policy as you've suggested seems like
> a can of worms to me.
>
> I believe the decision as to how a given command is to be terminated is
> best left to the implmentation. ie. Let sleeping dogs lie.
>

T 055 Page 73, Section 9.1

The SCSI Command Request may be excessively restrictive by defining a closed-ended list of request parameters. Is it implicit that a protocol private input and output parameter list is allowed, or should it be explicitly allowed.

This is also a valid question for the indication, response, and confirmation paramters.

>
> The goal of the architectural model is to specify behavior in a way
> that's protocol-independant. Adding protocol- or implementation-specific
> parameters in the manner suggested above will seriously compromise
> that goal and make it impossible to write portable software or
> firmware.
>
> In my opinion, compromising portability will seriously jeopardize the
> industry's significant software and firmware investment. Failure to
> protect that investment will discourage migration to newer technologies.
>

T 056 Page 73, Section 9.1

I would have expected the Data In Buffer Pointer and the Autosense Buffer Pointer to be inputs to the requests, since they are pre-allocated areas that are being made known to

the executor of the services at both ends of the link.

>
> The noation needs to be modified to identify pointer variables which receive
> outputs from a remote procedure call
>

T 057 Page 73, Section 9.1

The autosense data returned in the area defined by the Autosense Buffer Pointer may include protocol dependent wrapper information and status in addition to the sense data.

>
> Allowing for such implementation-specific "wrapper" information
> is outside the scope of SAM and is not needed to adequately
> describe essential, protocol-independant behavior.
>
> See also the response to comment 055.
>

T 058 Page 75, Section 9.2

The application client's buffer is not actually "a single, physically contiguous block of memory". It is actually a contiguous block of virtual memory or a logically contiguous block of memory.

>
> The buffer will be defined as a "logically contiguous" block of
> memory.
>

T 059 Page 75, Section 9.2

The device server is described as "must have the ability to transfer such data in increments smaller" than the total extent to be transferred. In fact, there is no such requirement, but should be reworded to indicate:

"The model also requires that the protocol be capable of transferring data in increments smaller than the extent specified in the command descriptor. The device server may have the requirement to transfer data in such smaller

increments."

>
> Agreed. The change will be made as specified above.
>

T 060 Page 75, Section 9.2

In the definition of Command Byte Count, it should additionally be clarified that verification of complete and correct information transfer may be impossible when data transfer segments overlap. A protocol should be allowed to prohibit overlapped transfers.

>
> Agreed. As discussed during the working group review, SAM
> will also be modified to recommend that device servers wishing to
> retain portability, not attempt such overlapped transfers
>

E 061 Page 76, Section 9.2.1

Third line, missing comma.

T 062 Page 76, Section 9.2.1

The last sentence of the section is incorrect as written. The initiator should assure that the the blocks of data are written into the buffer at the correct displacement within the buffer, regardless of the order in which the blocks were actually presented to the interface.

>
> No, the data must be written in the order received. Otherwise, an
> overlapping transfer that's written out of order will result in
> corrupted data.
>

T 063 Page 76, Section 9.2.2

The Device Server Buffer Pointer is managed by the Device Server making the request and is not necessarily an input or output parameter of the request.

>
> As noted above, the notation must identify caller-supplied pointers
> that reference input or output data.
>

T 064 Page 76, Sections 9.2.1 and 9.2.2

The transfers should not be considered successful unless the number of bytes transferred exactly matches the requested byte count. In the case of a write, an incomplete byte count can only occur in the block transferring the highest offset byte of data and indicates an incorrect length, which must be posted back to the SCSI initiator according to the rules for incorrect length. In the case of a read, such a mismatch in byte count is an error in the transfer process.

>
> The data transfer services always move the number of bytes
> requested. In the case you mention, the device
> server would invoke the service with an appropriately adjusted
> byte count then terminate the request appropriately.
>
> Failure to transfer the specified number of bytes is a fatal
> error within the service delivery subsystem.
>

T 065 Page 77, Section 9.3

This section appears to be redundant with section 8. See problem 15.

The "result" parameter codes disagree with those defined in Page 68, section 8.

>
> Any discrepancies will be corrected in the reorganized version.
>

T 066 Page 78-81, Annex A

This section should be removed from the standard. It offers nothing to the definitions of the standard.

It does, however, point out that there is some information missing in the definition of task set. I suggest that the

information from Table 1 of annex C be included in the normative document, probably in section 4.7.3.2 or section 7.

>
> While the information in the body of the standard must agree with the
> technical requirements set forth in the queuing model, the editor must
> have the latitude to select the clearest method of presentation.
>
> In that regard, I consider table 1 as illustrative material that
> supplements normative behavior described elsewhere.
> In my opinion, therefore, inclusion of table 1 is subject to editorial
> discretion.
>

E 067 Page 82-85, Annex B

This section should be removed from the standard. Subsequent updates should be documented with an appropriate cover letter to the draft standard and not within the standard document itself.

T 068 Page 86-94, Annex C
(Modification required to change vote to affirmative).

This section is redundant with and disagrees with the normative standard. It must be removed from the standard. It does not contain the necessary information about the management of tasks during the various states, information which is correctly included in the draft.

The following problems indicate information discrepancies and describe an appropriate correction for each discrepancy.

T 069 Annex C

The description of the ACA control bit in Annex C is incomplete and has been correctly supplanted by the normative information contained in the body of the draft.

T 070 Page 87, Annex C, section 1.1

"The target shall manage....all other task sets." is the information that should be transferred to section 4.7.3.2 as described in problem 066

T 071 Page 87, Annex C, section 2.1

This section is not correct and is supplanted by the draft.

T 072 Page 88, Annex C, section 2.1.4

"If the target accepts ... complete the current task." is correct information not presently explicit in the body of the draft. It should be added to section 8.4.

>
> As I interpret this text, a CLEAR AUTO CONTINGENT ALLEGIANCE issued
> with no ACA in effect is a NOP.
>
> I believe the description in clause 8.4 on page 72 covers this case.
>
>

T 073 Page 89, Annex C, section 2.2

The list of task terminations may be an appropriate addition to section 7. Section 7 is a little vague about the mechanisms by which a task is ended.

>
> Agreed.
>

T 074 Page 89, Annex C, section 3.1

The second paragraph indicates that more than one current task may exist at a time. This really means that more than one task may be enabled (and therefore in some state of execution) at one time.

The third paragraph indicates that more than one task may be actrive on a physical transport system at once.

While implicit in the body of the draft, it might be helpful to make these points explicit, possibly in section 7.3.

>
> Agree.
>

T 075 Page 91, Annex C, Table 1

The information contained in this table should be included in section 4.7.3.2 or section 7 as proposed in problem 66.

>
> Agreed.
>

From: Charles Monia
SAM Technical Editor

To: Members of X3T9.2

Subject: Proposed responses to Western Digital Comments on SAM Revision 12

The following are my proposed responses to the comments received from Western Digital on SAM revision 12. These were prepared following a number of email exchanges and, hopefully, accurately reflect the current state of all the issues that were raised.

The original comments are included verbatim followed by the proposed reply. The reply is preceded by a right arrow ">". Editorial comments to be incorporated as written are not accompanied by a reply.

Begin response
==========

pg 11, 3.1: You define request, response, and indication, but there is no definition for 'confirmation'. You also define lower level protocol without defining upper level protocol.

>
> The appropriate definitions and clarification will be added.
>

pg 11, 3.1.11: "...has than has..."

pg 14, 3.1.61: You shouldn't need a definition of SIP. Likewise for 3.1.63. Delete them. In any case, the definition given for SIP is true for ANY protocol!

>
> The definition fr SIP will be deleted.
>

pg 21, 4.2, 2nd line: The "...single request-response transaction." should be given an adjective like "virtual" or "effective", to differentiate from the "actual" transaction that passes through the service delivery subsystem.

>
> Agreed.
>

pg 22, Figure 2: This figure is good, in fact it is the cornerstone for the rest of the document. As such, certain conventions should be used within the figure which are then used in all subsequent figures. For instance, the exact same dashed line style for the "virtual" server request/response transaction should be used in all subsequent figures. I note that subsequent figures use different line styles, which made it difficult to follow what was really being passed to whom.

>
> Agreed.
>

pg 33, obj. def. 4: Hard to see how an initiator can exist without at least 1 Application Client. Also, the definition of Initiator Identifier (in words) says that the "content" of the object is protocol specific, yet the formal definition above says that it contains a value between 0..2^64-1. Which is it, or did you mean "encoding" as was said for the Target Identifier on the next page?

>
> The encoding of the initiator identifier is protocol-specific.
>
> SAM should point out that application clients are created to issue commands
> or task management functions and cease to exist once the operation
> terminates. Hence, if there are no tasks or pending task management
> functions, the number of application clients is 0.
>
>

Also (more importantly) all of the tools and services provided by SAM seem to allow for multiple Initiator and Target Identifiers only on the most simplified level: you can have more than one, but you can't relate one to the another within the same device. Given this, why don't we say:

Initiator = 1{Appl Client} + 1{Initiator Identifier}1

Likewise for target... The alternative is adding considerable complexity to make multiple IDs fully functional.

>

> Multiple I/D's are supposed to be aliases representing the same physical
> device.
>
> Since this was added by request, I'd like some feedback from the
> working group before I modify the definition. In my opinion, the
> issue of multiple target identifiers is irrelevant to SAM.
> Although the behavioral model requires one unique identifier,
> a system could implement more than one without violating any
> architectural requirements.
>
>

Another point: Where does it say that the ID may be less than 2^64 for a given protocol (like, say, SPI/SIP)?

>
> See clause 3.4.2 on pp 19. This information will be moved to clause
> 4.4.
>

pg 34, 4.7.2: Why even mention encoding for Target ID?

>
> Agreed. The last sentence of the target identifier object description
> will be deleted.
>

pg 35, 4.7.3: A consequence of having more than one Target ID is that a logical unit can have more than one Logical Unit Identifier....

>
> See response to comments on pp 33, object definition 4.
>
> As noted in that case, multiple identifiers for the same logical unit,
> can be considered as aliases for the same phyisical entity, and therefore
> have no bearing on the architecture.
>
> I suggest leaving this item open for discussion at the first meeting of
> X3T10.
>

pg 35, 4.7.3.1: "should" should be "shall", shouldn't it?

>
> The present wording was added by request to eliminate what was thought
> to be an unnecessary behavioral restriction.
>
> I suggest reopening the issue at the upcoming meeting of X3T10.

==========

pg 35-36, 4.7.3.x: The Device Server contains the Task Set (never mind George's objections regarding the Queuing Model, which I have come to agree with). From an object-oriented point of view, it seems that the Task Set and Task Manager need to at least be peers to allow the Task Manager to operate on the Task Set. With the Task Set being within the Device Server, I don't see how the Task Manager can do that without directly interacting with the Device Server.

So, we open a can of worms, between this and the Queuing Model. It seems that the hierarchy is at fault, not the objects themselves. So:

- Put the Task Manager, Target ID, and Logical Unit(s), under the Target.

- Put the Task Set under the Task Manager (keep Task under Task Set). This allows the logical units to interact with the Task manager as peers, without actually touching the Task Set. This also hides the Task Set organization from the rest of the model (as per the Queuing Model in Annex C).

- Logical Unit now has Device Server and LUN below it. Device Server should be:

Device Server = 1{Device Impl.}

- Another argument in favor of a single Task Manager per Target is that the Task Manager receives Task management requests like "TARGET RESET". Clearly a target level function. All other requests operate directly on the Task Set. QED.

>
> The model will be modified to define a single task manager per
> target. In addition, the task set definition will be specified such
> that it can easily be decoupled from the logical unit.
>

==========

pg 37, obj def 8: Under "Tag", it says: "An initiator shall not assign the same tag...". If you allow more than one ID per initiator, then the constraint is by ID, not by initiator. No change necessary if initiators can have only one ID.

>
> The text will be modified to require that the task identifier
> be unique.
>

pg 37, last sentence: The word "automatically" is not necessary.

>
> Agreed.
>

pg 38, 6.0: SOMEWHERE, it should say that Execute Command is a virtual request of some sort (a ULP transaction as back on Fig 5). This is presented as something real and is terribly confusing. Perhaps getting rid of it (at least as a formal entity) is best.

>
> Section 4.2 states that services are modelled as procedure calls.
> In the case of distributed services, these are remote procedure calls.
> The description should reiterate that the procedural model is a
> convention for specifying behavior.
>
> As you suggested in a later note, I will provide diagrams using
> conventions for showing multi-layer protocols, like figure 3-12
> in rev 6.5 of the P1394 spec.
>

pg 39, Task attribute: These are defined nowhere. If you follow the references, you get into a loop back to Obj Def 8, where they are listed and not defined.

>
> Agreed. This will be corrected.

>

pg 39, pointers: How can the server assign the data in and autosense pointers in the client? These should be passed from the client as places for the server to put them.

>
> Agreed. The notation should identify pointers supplied by the caller, which contain caller-supplied input data or receive outputs from the called procedure.
>

pg 39, Data-in pointer: The contents of the buffer can be perfectly well defined for other status values, like CHECK or TERMINATED.

>
> The specification should state that the application client shall consider the contents to be undefined unless supplementary information returned by the logical unit, such as sense data, indicates otherwise.
>
>

pg 39, command complete: same comment as for data-in pointer.

>
> Same reply as for data-in pointer.
>

pg 39, service response: I think the list is incomplete for linked commands. See later comment on pg 43.

>
> See reply below.
>

pg 39, autosense data: missing clause reference.

pg 40-41, tables 1-3: It seems inappropriate to show LBAs and such here. These belong in SBC. The tables here should be generic; say "Parameters" in each byte that is not otherwise occupied.

>
> Agreed. The figures will be removed to the SPC standard and replaced

> by a single generic table
>

pg 42, table 4: What possible use is a 16 byte CDB that has all bytes reserved for all commands sets? Either replace "Reserved" with "Parameters", or nuke them!

>
> See previous reply.
>

---

pg 43, 6.2.2: The notification when the flag bit is set should be part of the service response of the Execute Command. The service responses should be:

- command complete: The command has finished.

- linked command complete: The linked command has finished and the next command should be delivered via an Execute Command request.

- linked command complete with flag: The linked command (with flag bit set in the CDB) has finished and the next command should be delivered via an Execute Command request.

- etc.

These responses should be a piece of cake even for serial protocols, the easiest way being to copy the flag and link bits into the payload of the status packet returned.

>
> Agreed. The change will be made as specified above.
>

---

pg 44, INTERMEDIATE: The list of status codes at the end of the first sentence should include TASK SET FULL and BUSY, because these can occur on the first command of a link. Also, COMMAND TERMINATED could occur anytime by action of the initiator, which should break the link.

>
> SAM will, be modified as specified aboce.
>

pg 45, ACA ACTIVE: There should be a sentence for recommended initiator action, as there is for BUSY.

>
> Agreed.
>

pg 45, ACA ACTIVE, item c): "attribute" should be "Task Attribute". Confused me, anyway.

>
> The suggested change will be made.
>

pg 46, 6.4.2: Reference SBC in pgf 2.

pg 47, step 4: The "response" is not defined. I know what it SHOULD be! See pg 43 comment above.

>
> Agreed.
>

pg 47, general comment on linking: It is apparent that the application client is expected to issue another Execute Command for the next command in the link. I didn't see where it actually says this (what clause), or that the Task Identifier and Task Attribute must be the same for each command in the link.

>
> SAM will be corrected as specified above.
>

pg 48, 6.5.1: In the cold light of day, this seems to belong in SPC. READ CAPACITY?

>
> Agreed.
>

pg 48-49, 6.5.2: I like "faulting command"; makes it clear what you are talking about. May I suggest "faulted initiator" for the initiator that issued the faulting command? It isn't always clear which initiator you are talking about. For example, pgf. 2 under 6.5.2.1.

>
> Agreed.
>

pg 49, 6.5.2.1, item b). Criteria a) under 6.5.2.2 does not apply to a new command. Be more specific. This clause needs lotsa work.

>
> Agreed.
>

pg 50, pgf. at top beginning "When an auto...": Both sentences seem to say the same thing, sort of; i.e., the ACA flag of the first CDB that caused the ACA remains in effect for the duration of the ACA, even if you get more ACAs during the current ACA. Or something like that. Or did I miss something?

>
> The paragraph in question will be reworked.
>

pg 50, 6.5.3, 2nd pgf.: I don't understand the first sentence. Why not always detect overlaps? Give a reason in a note or delete the protocol specific provision and make it global.

>
> The reason for making it a protocol-specified requirement is due to
> the large tag space for protocols like FCP [I should have
> added "and hence the large perceived overhead to search for duplicate
> identifiers". In any event, I believe this is another item for further
> discussion].
>
>
pg 50, 6.5.4, item b): "unti"

pg 51, 6.5.6, 1st sentence: Just say sense data is made available when an ACA occurs.

pg 51, 6.5.6, 2nd sentence: "created" not "returned". Returning is done via REQUEST SENSE, etc.

pg 52, 6.5.7, 2nd sentence: Remove "automatically". Can be confused with autosense.

pg 53, 6.5.8: The items on clearing sense data belong under 6.5.6.

>
> Agreed.
>

pg 53, 6.5.8: Nowhere does it say that autosense is optional for a protocol (like SIP/SPI).

>
> SAM will be modified to explicitly state that support for autosense is
> a protocol option.
>
> Note that even when the protocol supports autosense, implementation by the device
> is optional.
>

pg 53, 6.6, item h): This has been truncated relative to SCSI-2. As it appears, I should do unit attention whenever ANYTHING happens. The full SCSI-2 sentence should be retained: "Any other event occurs that requires the attention of the initiator."

>
> SAM will be corrected as specified above.
>

pg 56: I believe you meant: "... all tasks shall behave as if they became Enabled upon entering the Task Set."

>
> The wording will be changed to:
>
> "The task shall behave as if it did not exist prior to entering the
> enabled state".
>

pg 57-59, 7.3: I would like to propose using the convention used in

IEEE P1394 for describing states and transitions. Be glad to help generate the words for you. The nice thing about it is that each and every transition and state are fully defined, and you can find each in the text. Everything is mixed up here.

>
> Agreed.
>

pg 57, 7.3: HEAD OF QUEUE (HOQ) tasks enter enabled. Is the intent to have undefined re-ordering if several HOQs come in at once, since they will all be enabled? Or should there be a mechanism to put HOQs into dormant? Or have I stepped in something horrible that the Queue Model SSWG tiptoed around?

>
>
> The assumption is that tasks are entered in the task set
> one at a time (SAM will be revised to make this point explicit).
>
> As far as I can see, HOQ behavior agrees with the queuing model, which
> states that:
>
> 1. Tasks entered into the task set before an HOQ task may complete
> before the HOQ task completes.
>
> 2. An HOQ task shall complete before any subsequent SIMPLE or ORDERED
> task.
>
> According to rule 1, any previous task, including a previous HOQ task, may
> complete before the new HOQ task.
>
> Rule 2 implies that an HOQ task will only inhibit completion of subsequent
> SIMPLE or ORDERED tasks.
>
> The only interpretation consistent with both rules is that HOQ tasks are
> always enabled.
>
> SAM will be modified to state the above explicitly.
>

pg 59, last pgf.: Give the figure and table reference numbers. Easier to read.

pg 60: Table 10 is missing.

>
> See the version distributed with the last mailing
>

pg 67, 7.4.4: I don't see anything under the description of restrictive reordering that requires only one task enabled at a time, or am I (once again) missing something? If only one at a time is allowed, then HOQs must be able to start Dormant, since you can't bump the already enabled task.

>
> The material on restricted reordering is incorrect and will be
> deleted.
>

pg 68, 8.0: ABORT TASK SET, ABORT TASK, CLEAR AUTO..., and CLEAR TASK SET need Initiator Identifier as an argument. Also pages 70-71.

>
> The initiator identifier is an implicit argument in all remote
> procedure calls. Making it explicit in the calling sequence would
> incorrectly imply that it could be modified by the caller.
>
> Appropriate explanatory text will be added.
>

pg 68, 8.0: TARGET RESET argument should be Target Identifier. More could be sent in a given protocol, but that's all that is REQUIRED to do the job at the SAM level.

>
> SAM will be corrected as specified above.
>

pg 69: Add Initiator Identifier and Target Identifier to argument list.

pg 70, 8.2, last pgf.: "A response of Func. Comp. indicates that the task no longer exists." This is in conflict with your definition of Function Complete on page 68.

>

> The paragraph will be modified to correct the apparent conflict.
>

pg 70, 8.3: Replace "terminate" with "abort". Terminate has a different meaning, as in 8.7.

>
> Agreed.
>

pg 70, 8.4: The last sentence is incorrect. A service delivery failure is always possible. Delete the sentence.

>
> Agreed.
>
pg 71, first line: "may" should be "shall", maybe?

>
> Yes.
>

pg 71, 8.5: All occurances of "terminated" should be "aborted".

>
> SAM will be changed as specified above.
>

pg 72 on, 9.0: The text formatting makes it very unclear where things begin and end. Highlighting and formatting should be applied as in previous pages.

>
> Agreed.
>

pg 73, 9.1: It is just not clear what these are relative to Execute Command. I think what needs to be done is to duplicate Figure 5 here, showing each service and primitive on the figure.

>
> The presentation will be modified to show the distributed service
> being invoked followed by the related calls to the LLP protocol
> service layer.