Letter to APTA Consultants on Dialogs and Implementation

22 July 2002
John [Fayos],

I was tasked by the TCIP Technical Working Group to request your review of a white paper on the Use of Error Codes in TCIP Code Lists (see attached). The TWG is particularly interested in the impact the documented alternatives (or other alternatives) may have on the simulation you are building as well as on future dialog definitions.

We are trying to make a July 31 deadline for resolving all open issues. Your consideration of this matter prior to July 31 would be helpful.

Thank you.

Sincerely,

Polly Okunieff
43 Jamaica St.
Boston, MA 02130

tel: (617) 983-3364
fax: (617) 983-9827
email: okunieff@world.std.com

1. Response from Critical Link (APTA Simulation Consultant)

From Tim Slater, Critical Link

24 July 2002
Polly,

Our understanding of Option Three is as follows: In response to any request, the Requestor may either get a "message" containing the requested data or an "error message".

The error message you propose looks reasonable, but may need to be further defined/refined as part of the dialog effort. In particular, are the "reserved" codes reserved for standards definition, or are they reserved for user definition?

Also note that it would not be prudent to "insist" that an error message be sent - some applications may prefer no response, and there is also the issue of partially complete data. Another example is "Enumerated types" that may lend themselves to having imbedded error codes, so there could be some application specific ways for handling errors.

Having an error message defined in the standard should not preclude other mechanisms for handling error conditions.

If we understand Option three correctly, we think it is the best way to go.

-Tim

response from John Fayos, 25 July 2002

Polly:

Probably not much I can add at this point to what Tim and Rob have already said -- Tim was pretty clear about agreeing with "alternative #3" and I believe Rob favors the same. I also agree with the concept described in "alternative #3". Rob also went into more detail as far as his concerns down the line regarding the complexities and hazards in attempting to standardize "functional" dialogs involving specific error conditions -- similar to the concerns that I had shared with you in NYC back in June.

Thanks,

- John

2. Response from ARINC (APTA Consultant on TCIP Dialogs)

From Rob Ayers, ARINC

24 July 2002
Polly-

I looked at the paper & offer the following thoughts in support of the basic approach of using a single error notification reply message for all cases where the recipient of a TCIP message cannot process the received message:

1. Suggest that the approach of embedding error codes down in the data elements leads to too much detail in the specification & ends up leading us to specify internal behavior which does not require standardization & may inhibit vendor acceptance.

2. Suggest that the standard take the high road & limit its scope to message exchanges between systems. In this context error notification messages are only required for a limited number of messages (e.g when a query or subscription request is rejected). In these circumstances the message can contain an error code as its primary data element. If there is a desire to identify the guilty data element in the original message, an optional index to the faulty data element could be included in the error notification message, eliminating the need for a sophisticated identifier. Internal handling & recovery from the error can remain vendor defined.

3. This approach also limits the type & complexity of the error codes themselves. For example, we do not need a divide by zero error code as division does not occur in the message.

4. This also eliminates the need for separate error processing dialogs. If each dialog pattern terminates in a safe state in the presence of an error, there is no need to specify additional error recovery software. My experience is that the specification and implementation of such

error recovery schemes is very costly, and in many cases more costly than implementation of the transaction (dialog) that actually performs useful work. My observation is that current interfaces between working systems from different vendors operate without sophisticated error recovery schemes, primarily because the costs of such mechanisms outweigh the benefits.

5. This approach also lends itself to reuse in the case where a query or subscription request cannot be processed because of a security concern (e.g. unauthorized requester). Security error codes can simply be added to the list of standard error codes.

In general I think it should be a goal to keep the dialogs as simple and understandable as possible. Every increment of complexity we add will add cost, and inhibit adoption by the suppliers.

Further note- There should be a rule that you cannot send an error notification message to notify another entity of a defective error notification message (to prevent deadly embrace).

Hope this is useful.

Rob