

From: [David RR Webber \(XML\)](#)
To: [ebXML MSG;](#)
CC:
Subject: [ebxml-msg] [FWD: [ebxml-msg-comment] Public Comment]
Date: Sunday, July 02, 2006 7:28:15 AM
Attachments:

By a miracle I did not think this was SPAM.

Can someone please respond to Ronald on this?

Thanks, DW

----- Original Message -----

Subject: [ebxml-msg-comment] Public Comment

From: comment-form@oasis-open.org

Date: Sat, July 01, 2006 6:45 pm

To: ebxml-msg-comment@lists.oasis-open.org

Comment from: ronald@vankuijk.net

Name: Ronald van Kuijk

Title: IT Architect

Organization: Private interest

Regarding Specification: ebMS 3.0 Public Review Draft 1

A few things that caught my attention trying to use the xsd with xmlbeans and reading the docs:

- The response example message in 5.3.1 has the collaborationInfo in the wrong place (line 1520-1524)
- the mustunderstand and version on the message element are in the same namespace according to the xsd but in different ones according to the docs
- in the ebMS Header example (line 1151) contains a PartInfo element with a scheme and description element. These are *not* in the schema in the same document. In 5.2.1.13 there is information that it is allowed

to use these elements in the same namespace

- in the example in 5.2.1.3 and the docs of 5.2.1.4 there is a reference to a type attribute. This attribute is not in the docs (PartyId is not an xmltype)

To unsubscribe, e-mail: ebxml-msg-comment-unsubscribe@lists.oasis-open.org

For additional commands, e-mail: ebxml-msg-comment-help@lists.oasis-open.org

To unsubscribe from this mail list, you must leave the OASIS TC that generates this mail. You may a link to this group and all your TCs in OASIS at: https://www.oasis-open.org/apps/org/workgroup/portal/my_workgroups.php

From: [David RR Webber \(XML\)](#)
To: [ebXML MSG;](#)
CC:
Subject: [ebxml-msg] [FWD: [ebxml-msg-comment] Public Comment]
Date: Sunday, July 02, 2006 7:28:38 AM
Attachments:

Can someone please respond to Ronald here.

Thanks, DW

----- Original Message -----

Subject: [ebxml-msg-comment] Public Comment

From: comment-form@oasis-open.org

Date: Sat, July 01, 2006 9:27 pm

To: ebxml-msg-comment@lists.oasis-open.org

Comment from: ronald@vankuijk.net

Name: Ronald van Kuijk

Title: IT Architect

Organization: Private interest

Regarding Specification: Public Review Draft 1 ebMS3.0

The xsd in the doc contains nothing about the eb:Messaging/eb:

UserMessage/eb:MessageProperties element

The eb:Messaging/eb:UserMessage/eb:PayloadInfo element is required according to the xsd, but in 4.3 line 866 it says the element is absent in the "Default P-Mode.businessCollaboration"

The 'default' values in 4.3 all end with a period. These should not be a part of the value and thus removed

To unsubscribe, e-mail: ebxml-msg-comment-unsubscribe@lists.oasis-open.org

For additional commands, e-mail: ebxml-msg-comment-help@lists.oasis-open.org

----- To unsubscribe from this mail list, you must leave the OASIS TC that generates this mail. You may a link to this group and all your TCs in OASIS at: https://www.oasis-open.org/apps/org/workgroup/portal/my_workgroups.php

From: [David RR Webber \(XML\)](#)
To: [ebXML MSG;](#)
CC:
Subject: [ebxml-msg] [FWD: [ebxml-msg-comment] comment from Pim van der Eijk
Date: Thursday, July 06, 2006 6:58:51 AM
Attachments:

FYI, DW

----- Original Message -----

Subject: [ebxml-msg-comment] Public Comment
From: comment-form@oasis-open.org
Date: Thu, July 06, 2006 9:13 am
To: ebxml-msg-comment@lists.oasis-open.org

Comment from: lists@sonnenglanz.net

Name: Pim van der Eijk
Organization: Sonnenglanz Consulting BV
Regarding Specification: EbXML Messaging Services 3.0, Part 1, Core Features

Here is some feedback on the ebMS3 specification.
I hope you find it useful.

Kind regards,

Pim van der Eijk

Line 210-214: add note that also multiple payloads, possibly of different MIME types, can be transported in a single ebMS message (important advantage in some applications)

Line 290

Change:

"/x:MyHeader/x:SomeProperty/@value1"

To

"/x:MyHeader/x:SomeProperty/@attribute1"

Line 294:

The prefix "wssswa" is defined in

<http://www.oasis-open.org/committees/download.php/16672/wss-v1.1-spec-os-SwAProfile.pdf>

with the value

"http://docs.oasis-open.org/wss/oasis-wss-SwAProfile-1.1.xsd"

rather than

"http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0.xsd"

(The URL doesn't resolve to an actual XSD at the doc.oasis-open site though).

Line 309:

These definitions seem to assume that the terms "Producer" or a "Consumer" are reserved for Endpoints and cannot be used for Intermediaries. Is that correct?

Line 322:

"There are two types of ebMS Messages:"

Are these types exclusive? If a message has both eb:SignalMessage and eb:UserMessage elements, what is its type? A pull request could at the same time push a UserMessage. Is that allowed?

Line 323:

The definition does not say which entity initiates a Signal Message, whereas the definition of User Message does.

Line 342:

"receive and process error messages associated"

should this be:

"receive and process Signal Messages associated"?

Line 352:

"enough message data"

->

"enough message (meta)data"?

(some of the data needed may not end up in the actual message)

Line 382-383:

Formatting (no new paragraph)

Line 364:

Section 2.2

Line 489:

"responses to these." -> "responses to these requests."

Line 495, Message Pulling:

It looks like this mechanism can be used to select a particular mailbox.

Could this be generalized to a mechanism where the Initiator wants to retrieve a response to a particular message? The Responder could have a database of RefToMessageId values of messages waiting to be pulled.

The requested message may not be the first in the queue.

Could the Initiator provide additional filtering information, e.g. only retrieve message less than 1 MB in size? This would increase the value for SMEs that have limited bandwidth connections.

Or should the Initiator/Responder negotiate policies about which message to put in which Message Partition based on criteria like size?

Is there only one message being pulled for each PullRequest / MPF?

Line 509:

Uses the "P-Mode" concept that hasn't been introduced at this point.

Line 527:

Is there no From/To PartyInfo in a Pull Request?

How does the Responder know who is pulling?

An initiating MSH may operate on behalf of multiple Parties (or a single Party with multiple IDs). Is there a need to be able to specify a

"To" (pulling messages sent "to").

Or is an MPF always unique for a particular PartyId ?

Line 533 and 548:

While it is an non-normative example, it may be clearer to change these to something like:

```
"<eb:MessageId>UUID-4@receiver.com</eb:MessageId>"
```

```
"<eb:MessageId>UUID-5@sender.com</eb:MessageId>
```

```
<eb:RefToMessageId>UUID-4@receiver.com</eb:RefToMessageId>"
```

Line 640, section 3.4.1:

It is not clear from this text (and earlier, line 479-480), whether the concept of MPFs makes sense outside the context of Pulling. When A pushes messages of two types to B, is there any benefit in using MPFs?

Line 681, no distinct addressing for MPFs.

Priority handling (other than queues at sender and receiver) often also depends on networking management equipment with bandwidth management (packet shapers). Those products classify network traffic based on source/target IP addresses, URIs, port numbers. So some mechanism to associate MPFs with distinct addresses (statically or dynamically) would be useful.

Line 786:

"the the ebXML Application." -> "the ebXML Application."

Line 887:

"Because either packaging option can be used, implementations MUST support non-multipart messages."

Should this be:

"Because either packaging option can be used, implementations MUST support both multipart and non-multipart messages." ?

Line 901:

"parts. containing additional" -> "parts containing additional"

Line 903:

"requires an XML content." -> "requires XML content."

Line 918:

"Package contain a type" -> "Package contains a type"

Line 965:

"Namespace declaration" -> "A namespace declaration"

Line 1030:

"the namespace for" -> "the namespace prefix for"

Line 1079:

"there MUST NOT be additional Payload Containers."

Does this mean the ebMS 3.0 specification precludes composition with other SOAP protocols that want to attach data? (Somewhat theoretical question, admittedly).

Line 1119 and 1148:

"Both eb:UserMessage element and eb:SignalMessage element MAY"

Does this mean

"Both a eb:UserMessage element and a eb:SignalMessage element MAY"

or

"Both eb:UserMessage elements and eb:SignalMessage elements MAY" ?

If the spec allowed multiple eb:UserMessages, it could easily support batch/unbatch operations.

Line 1159 and 1165:

Not sure why "(no change beside renaming)" is there.

Line 1192:

"cid:foo", note that "foo" has to be a MIME content ID, i.e.

"foo@example.com".

Line 1225: this allows for payloads not in the message, but line 1386 doesn't.

Line 1230:

Why is the element "timestamp" in lower case? eb:Timestamp would be more consistent?

Line 1253:

What is the semantics of multiple PartyId elements in a From (or To)? EbMS requires these to be alternative names for a single organizations. Has this requirement be dropped in ebMS 3? If yes, that should be specified explicitly.

Note: various people have struggled to squeeze hierarchical addresses (Company/ Division/ Unit/ Employee) in the ebXML PartyId field.

Line 1311:

"If a CPA is referred to and a receiver determines that a message is in conflict with the referred CPA, the appropriate handling of this conflict is undefined by this specification."

What does this mean for ebCPPA 3.0? Should it have a normative ebMS3 binding that defines how this is handled in an interoperable way?

Line 1320:

"a profiling exercise" -> "profiles using this specification"

Line 1382:

"(see Section , for details)" missing reference.

Line 1386:

"The PartInfo element is used to reference either a MIME attachment or an XML element within the SOAP Body, according to the value of the href attribute, described below." Line 1215 mentions URL-identified payloads too. Does the spec allow this or not? Not according to line 1391-1393.

Note: there may be a case for disallowing URI payloads, i.e. requiring all payloads to be in the message.

The semantics of a URI in a PayloadInfo not in the MIME envelope is unidentified. Should the receiving MSH attempt to download the payload from the Internet? What if it can't access it?

More useful would be some sort of ebXML Signal or extension protocol

for downloading/pulling large attachments separately from the main message is useful. E.g. if there are two attachments, a 40 KB XML document and a 160 MB PDF file, it would be nice if the first can be pushed and the second pulled on demand and based on some combination of MessageId and Content-Id.

Line 1405: Schema element.

Does this mean that the receiving MSH must validate the payload?
If not, why does the message handler need to carry this information?

Line 1415: Instead of Description, there are use cases for allowing extensible metadata about payloads. E.g. to get the MIME type, you now have to find the MIME part. Sometimes it makes sense to have that in the PayloadInfo. If the attached payload is XML Encrypted, PayloadInfo could reveal the "real" datatype. If the payload is a TIFF image, the PayloadInfo could provide meta-data.

Line 1425:

eb:version="2.0" ?

Line 1452:

"However, ebMS message can" -> "However, ebMS messaging can"

Line 1650:

Is this list exhaustive? If ebMS is composed with other SOAP protocols, can they be escalated too?

Line 1669:

"reported in a log I" -> "reported in a log in"

Line 1830-1833.

How can a containing element eb:Messaging be encrypted while a contained element eb:PartyInfo is not? An example would help here ..

Line 2406-2423, section 8.2.3:

If both security and reliability are used, there is presumably a required processing order of the security and reliability headers? Where is this specified?

Line 2448-2452:

The semantics of ack was NOT clearly a delivery semantics in ebMS2. The ebMS3 specification is more precise in defining At-Least-Once as a delivery contract. But the specification still allows the option of acknowledging a message that is not delivered, e.g. in case of a packaging error that is detected after the reliability processing of a received message, where only the generation of an error is required. Shouldn't the spec tighten-up the requirements for acks, or at least provide a conformance profile that requires a tighter semantics? Should this type of difference be documented more clearly? Is there a contract P-Mode property that allows partners to specify this level semantics of Reliability (including or not successful "Deliver") ?

When RM-Deliver is executed before Deliver, there may be issues in the case of failover. Suppose the MSH system (or the interface from the MSH to the enterprise system) crashes, and processing is transferred to a standby system, that system may want to have all messages delivered in some previous time interval (e.g. 15 minutes if the standby system needs a few minutes to come to service) redelivered, even though they have already been acknowledged. This in order to get any messages possibly stuck between the MSH and the enterprise system before the crash.

Is this supported by either the WS-R or WS-RM specifications? If not, could ebMS3 add some support here?

Line 2499-2502:

It is worth pointing out that the order of RM-Deliver and Deliver could be different.

Line 2637, Section 10:

There are some constraints between Reliable Messaging and ebXML processing.

E.g. all messages in a WSRM sequence should be in the same ebXML conversation and CPAId.

Where are these constraints specified?

How are they enforced in an implementation if it uses non-ebXML aware generic RM processors?

There is now no ebXML concept of Message Ordering.Can

organizations still use the ordering

features of the underlying RM specifications?

E.g. certain service/action/cpaId/conversationId initiate new sequences, and others terminate them. Is there a way to specify this correlation.

Partners may want to specify that certain Service/Action combinations, with a new ConversationID and in the same CpaId, are to be performed in-order. How do they specify this, and how does the ebMS processor know about this?

Line 2730:

"This become a" -> "This becomes a"

Line 2851 and 2858:

These examples have invalid Content-Ids (no "@" sign).

General comments:

SyncReply module in ebMS2

A mapping from the CPA properties for SyncReply to P-Modes would be useful, for the existing ebMS2 users looking at ebMS3.

EbMS2 has a Message Status service.

This seems to have disappeared?

Was it considered not useful? It seems quite relevant in multi-hop. Do any of the underlying specifications provide this functionality?

EbMS2 has a Ping service.

What happened to it?

EbMS2 had a section C Supported Profiling Services.

It would be quite useful to have a similar table that maps these profiles to ebMS3 with WS Security.

EbMS2 had a section 7.6 with reliable messaging combinations.

When Part II adds support for multi-hop, it would be useful to review how this could be mapped to ebMS3 with WS-R or WS-RM.

To unsubscribe, e-mail: ebxml-msg-comment-unsubscribe@lists.oasis-open.org

For additional commands, e-mail: ebxml-msg-comment-help@lists.oasis-open.org

----- To unsubscribe from this mail list, you must leave the OASIS TC that generates this mail. You may a link to this group and all your TCs in OASIS at: https://www.oasis-open.org/apps/org/workgroup/portal/my_workgroups.php

From: [David RR Webber \(XML\)](#)
To: [Gait Boxman](#);
CC: ebxml-msg-comment@lists.oasis-open.org; [ebXML msg](#);
Subject: [ebxml-msg] RE: [ebxml-msg-comment] Public Comment
Date: Monday, July 10, 2006 11:31:41 AM
Attachments:

Gait,

Thanks for the comments. I've cc:d the main list so others can comment here.

The one item I'm familiar with is the "stage delivery" support for push/pull.

I agree that you can do this today with v2 - but its not directly supported - so implementations are not consistent. Therefore the need is to provide a formal mechanism in v3 here.

Again - application practice has shown we do need this - particularly for large payloads - or to avoid peak surges overwhleming central systems - so non-time-critical deliveries can be picked up at times when the receiver has slack capacity. As you note it does require longer message persistence by the sender - but again - this can be an essential part of the process - e.g. the sender has fulfilled legal requirement to submit by a deadline - and locked their submission package - but the actual payload delivery can be deferred. Clearly you have to have agreement on the CPA for the staged delivery responsibilities and timings. I believe there is enough parameters here to accomplish those settings between two partners so timings and delivery responsibility is not an issue. As always - probably some example would help clarify the exact details.

That is just one use case from my experience - I know others have more too.

Anyway - the bottom line is that real world applications have shown us the need to have support in v3 for this formal push/pull mechanism.
Hope that is enough clarification on the rationale behind it.

Thanks, DW

----- Original Message -----

Subject: [ebxml-msg-comment] Public Comment

From: Gait Boxman <gait.boxman@tie.nl>

Date: Mon, July 10, 2006 5:13 am

To: ebxml-msg-comment@lists.oasis-open.org

Please find below some concerns on the draft ebMS 3 specification.

Unfortunately I was not able to go into detail, and can only comment on some high-level issues based on a feature list from Jacques.

- Section 3 introduces a 'message pulling' feature on the SOAP level to overcome limitations such as availability of static IP

I find it strange that one would need such a feature, since ebMS 2 already provides SMTP/POP3 based transport which already covers such a

need. The main issue with this is shifting responsibilities. A sender is now required to keep the message available for the recipient on his server, rather than dropping it in the realm of the recipient when ready. This blurs the division of responsibility and leads to issues with message turnaround times.

- Section 2.2 introduces 'message exchange patterns', which attempt to tightly couple business process with a particular message exchange.

This

shouldn't be part of ebMS as it introduces too much dependency between

process and messaging. The problems with sync and async messaging modules wrt to MEP already indicate you're not on the right path here.

Please stick to the reftomessageid for linking messages and allow business process to design the message patterns for the process. All we need to do here is making sure people 'can' relate messages if they need to.

- Section 5,7,8 - changing the header formats to comply with WS*.

The main issue is backwards compatibility. While I learned from Jacques

that SwA is still the way to go, moving critical information about in the SOAP Envelope will require recoding at the core. This will hamper

migration from ebMS 2 to ebMS 3 environments, if not block them completely, leaving the communities on an island and requiring implementers to maintain two versions.

- Processing Modes.

Having a concise set of data for processing modes is a good idea. In the past, we've seen people struggle with CPA's to configure their MSH's. Whether or not we need a 'formal' PM document is another issue, I believe the content is already inside the ebMS2 spec, it just needs to be elevated into a concise document essentially detailing the various parameters that one may set.

- Conformance Profiles.

Using a conformance profiles document separate from the main spec or embedding it doesn't make a lot of difference. However, allowing choice

on things like the version of SOAP or WSS introduces options that may hamper interoperability on the larger scale. IMO, It would be in the interest of the market (both developers and users) to fix versions of underlying protocols as much as possible in order to avoid flexibilities that may divide the market. Even when e.g. SOAP 1.2 is backwards compatible with SOAP 1.1, it is better to decide on SOAP 1.1 or SOAP 1.2, rather than leaving it open. It simply reduces the number of *possible* interoperability issues and the amount of test sets we need to add for interoperability.

If ebMS 3 doesn't decide on SOAP 1.1 or 1.2, it probably also doesn't decide pro or against SOAP 1.3, which may be not so backwards compatible. This extends to all the underlying protocols, and the combinatorial exponent of them..

HTH, kind regards.
Gait Boxman.

To unsubscribe, e-mail: ebxml-msg-comment-unsubscribe@lists.oasis-open.org

For additional commands, e-mail: ebxml-msg-comment-help@lists.oasis-open.org

----- To unsubscribe from
this mail list, you must leave the OASIS TC that generates this mail. You may a
link to this group and all your TCs in OASIS at: [https://www.oasis-open.org/apps/
org/workgroup/portal/my_workgroups.php](https://www.oasis-open.org/apps/org/workgroup/portal/my_workgroups.php)

From: [David RR Webber \(XML\)](#)
To: [ebXML msg;](#)
CC:
Subject: [ebxml-msg] [FWD: [ebxml-msg-comment] Public Comment from Yangsun Park, KorBIT
Date: Tuesday, July 11, 2006 9:38:50 AM
Attachments:

FYI, DW

----- Original Message -----

Subject: [ebxml-msg-comment] Public Comment
From: comment-form@oasis-open.org
Date: Mon, July 10, 2006 10:54 pm
To: ebxml-msg-comment@lists.oasis-open.org

Comment from: ysplays@postech.ac.kr

Name: Yangsun Park
Organization: KorBIT
Regarding Specification: Public Review Draft 1 ebMS3.0

ebMS 3.0 is seem to get more flexibility to adopt other existing specifications than ebMS 2.0 specification. I understand P-Mode is on the context. MEP and MPF give basic view for connect messaging to business scenario and give solution for some technical problem. I give you brief comments on new concepts in ebMS 3.0, pulling, MEP, MPF, P-Mode.

I agree with that “pulling” is very useful for small businesses to operate ebMS endpoint especially under condition of dynamic IP addresses. But, some implementation issues are concerned. **The system may not handle the workload of pulling when messaging with many business partners or processing lots of messages. (This is out of the scope of this specification, but I think it’s worth to consider.)**

In the specification, there's no mention how to manage the pulling; how the system manages the messages when messages are already sent to MSH and the pulling request is delayed with known reason. For using MEP, the agreements between partners should be described in CPA, but there's nothing mentioned on the spec about it. It is recommended to suggest how business partners can use the MEP or to reference CPA specification at least.

In case of MPF, it is also needed to consider how to handle the unexpected situations. If using queue to implement the MPF system, there can happen problem when several business partners pull message from same MPF queue.

P-Mode is related to the implementation issues, but the configuration information seems to affect the business scenario. It seems to be natural the p-mode is integrated to the CPA.

The fact sheet is represented very well, and I think that it will be easier to understand if you mention the MEPs used in the specifications.

To unsubscribe, e-mail: ebxml-msg-comment-unsubscribe@lists.oasis-open.org
For additional commands, e-mail: ebxml-msg-comment-help@lists.oasis-open.org

----- To unsubscribe from

this mail list, you must leave the OASIS TC that generates this mail. You may a link to this group and all your TCs in OASIS at: https://www.oasis-open.org/apps/org/workgroup/portal/my_workgroups.php

From: [David RR Webber \(XML\)](#)
To: [ebXML msg;](#)
CC:
Subject: [ebxml-msg] [FWD: [ebxml-msg-comment] Public Comment]
Date: Wednesday, July 12, 2006 7:30:26 AM
Attachments:

----- Original Message -----

Subject: [ebxml-msg-comment] Public Comment
From: derrick.evans@bt.com
Date: Wed, July 12, 2006 9:04 am
To: <ebxml-msg-comment@lists.oasis-open.org>

Thanks for the opportunity to comment. Sorry this is late.

My comments on the ebXML ms 3.0 spec are as follows.

Section 2.2 MEPs

This section is useful but the problem is how far to go. The document mentions the use of eb:RefToMessageId but not conversationId and does not talk about the relation in time between the various messages.

So I am not sure what role this section plays in a document on the messaging standard. I suspect that ebBP or some such would be a better place to talk of these things in terms of the structure of transaction patterns?

Section 3 - Message Pulling

If the challenges can be met then I see this as a great step forward for ebXML ms3.0.

In our current communities we have been moving to ebXML ms2.0 from a

mechanism based on XML posted over http with responses being polled by the client.

The moment we explain that we are moving to push push many of our partners respond with "it wont work".

Our intention would not be to move to an exclusive push pull model but rather when outbound messages fail to be received we would take those messages and requeue them such that they could be pulled by a partner as part of their recovery process.

Section 4 - Processing Modes

I think this concept is a very good one. Not sure why the default P-Mode has no reliability. My view of ebXML ms is that a key feature is reliable messaging.

Conformance Profiles

Having choices is good however we are in a very diverse community (using ebMS solutions from three different vendors in the community plus an open source implementation and Three home grown ones).

When it came to interop even our use of ebXML CPA/CPD threw some of them.

It would be useful to try and come of the fence here even if it is to make a stand on the default first port of choice for all the commercial vendors.

Thanks for all the hard work on this standard

Derrick Evans
BT OneIT

----- To unsubscribe from
this mail list, you must leave the OASIS TC that generates this mail. You may a
link to this group and all your TCs in OASIS at: [https://www.oasis-open.org/apps/
org/workgroup/portal/my_workgroups.php](https://www.oasis-open.org/apps/org/workgroup/portal/my_workgroups.php)