

**** is used to highlight sections. Comments are mainly thoughts while reading through the spec.

General comments:

Is there room to separate (or separate even more) the following?:

x ebXML collaborative business process vs internal business process
x general negotiation vs ebXML CPA specific negotiation
x There is no schema for a CPA template. What is a valid CPA template? As once discussed a CPA only has one deliveryChannel element whereas there can be more in a CPA template to keep the preference ... Probably the CPPA schema is valid for a CPA template but ... not really ... Glossary says that a CPA template is a CPA with open fields, hence a CPA template will never be a valid CPA. A CPA template cannot be validated, can it? Maybe a CPA template does not have to be validated ...

*** Section 3.2

x Is it necessary to state that an negotiation algorithm is not included in version 1.0? Might otherwise someone think that this spec also provides those negotiation algorithms?

x Maintaining a large number of partner relationships. This negotiation scenario might have different/new negotiation requirements ... to be thought.

x "It will reduce the need for human intervention ..."? Does this sentence mean less human intervention or no human intervention? I think of the later.

*** Section 3.4

x Line 231, 239, 241, 245: Link to various file is invalid, but it says To Be Defined (TBD).

x Line 250: I think the CPA should have an attribute with possible values, such as "CPA template" or a "final CPA" or simply a "temporary-test-something-that-looks-like-a-CPA-and-might-become-a-CPA".

*** Section 3.6

Why is this necessary for people who have to create CPP's? The NDD info should actually be in the CPPA Spec ...

*** Section 4

x Line 279: "It does NOT define negotiation algorithms in detail". It rather does not at all define negotiation algorithms ... or later (line 281) it does not define negotiation strategies at all.

Maybe more on the difference between the negotiation algorithm and a negotiation strategy could be provided.

General: Will you put any negotiation algorithms into this specification? Then the negotiation algorithms will also be ... under

the IBM patent ... (not sure what the facts are here)

x Line 286: I thought the term "draft CPA" should no longer be used and the term "CPA template" should be used. Or a description of what the difference is, if there is any anymore. Might have to be aligned with the CPPA Spec and maybe the ebXML Architecture Spec. if not already done.

x Line 286: Section 6.2 only talks about CPA template, not one word about draft CPA.

x Line 294: A NCPA is for two specific parties, including their party info etc, maybe highlight that a NCPA is a sort of a skeleton CPA where the parties who want to negotiatiate just need to fill in the parties specific information (eg party id). I think this has to be covered somewhere ... like how the NCPA gets fully ready to be deployed...

x Line 303: "should invoke human intervention" ... how shall the system contact the negotiation operators? by sending an email to indicate that the negotiation failed?

x Line 218: it seems that rejecting an offer might become a tactical maneuver, especially the rejection of the initial offer with a new ndd... in case the receiving party just does not want to use the newly created CPA template and NDD... Maybe this cannot be circumvented.

*** Section 5

x Line 339: in 5 it said a CPA template and its NDD should be used for the CPA negotiation whereas in the figure also the CPP(A) and CPP(B) as well as NDD(A) and NDD(B) are shown to be an input to the CPA negotiation. Maybe have a figure showing the CPA template composition with the CPPs and NDDs as input and then the CPA negotiation.

I liked to call it the "CPA formation" which includes the CPA composition and the CPA negotiation.

Shall it be possible to run a CPA negotiation with 2 CPPs (eg no CPA template)? When I remember it also says somewhere that the original NDDs or CPPs might be used in the CPA negotiation to make sure that the initial preferences are respected during the CPA negotiation. ... in the end the CPA negotiation algorithm (as stated previously that they can be proprietary etc) can include whatever they like, because that is an internal business process ...

x Line 354: General: NDD: 1) The NDD for the CPP's have to be created (maybe manually), 2) NDD for the CPA template in the CPA composition has to be created (probably automatically). Maybe some mnore information about the NDDs and their creation is necessary...section 10 talks more about NDD's so that's OK.

x Line 383: Not sure if I understand the difference between a real and virtual NCPA. In both scenarios the missing party information is retrieved somehow.

x Line 415: The negotiation process diagram suggests that everything is part of the negotiation process, even the CPP, NDD formation ... The diagram is very helpful to show what is going on, how to get to a CPA from the beginning. If considering that the CPA formation process (CPA composition and CPA negotiation) is very important in adhoc e-biz, this diagram shows it very cleary how to get it but it also shows that it is not a straight forward process but can become tricky in the negotiation.

x Line 429: Prior to this point, there is a very important point in my opinion. The Initial Offer receiving party has to do a big job here: validating the cpa template and the ndd! For this process it sort of also has to create a CPA template and compare the two CPA templates if it can agree to what will be negotiated. The same for the NDD, it has to make sure it does not get cut short somewhere, that all its negotiation items are fairly represented in the ndd which goes with the new CPA template.

Here I see currently the biggest hurdle to get started with the negotiation as such.

x Line 438: Does the party fill in details such as its Party ID here or prior to the last counter offer? That could be done anytime during the negotiation, right?

x Line 459: Seems like the negotiation start without a CPA template but two CPPs are shown here whereas this should be done by the CPA composition ... matching of CPP's and to create a CPA template. This minimum level of matching might go into the CPA composition part.

x Line 500: Another service could be to provide "CPA composition" service where a party can select a potential partner CPP to compose a CPA template with its own CPP.

x Line 501: This might become a possible business scenario where a negotiation agent does negotiate on behalf of its clients (cpp providers).

*** Section 6

x Line 513: The problem with a CPA template is that it cannot be validated against a schema and a tool must be able to pick up what parts of a CPA template are not filled out yet. That's supposed to be the NDD. It would be nice to be able to validate a CPA template with its NDD (somehow overlaid) against the CPA schema.

x Line 508: Do you need to keep talking about a draft CPA after all?

x Line 533: Is it necessary to provide the Party ID? The CPA composition process should be able to pick the Party ID and put it into the CPA template. If there is only one matching transport element in both CPP's then the CPA composition process will pick that one and put it into the CPA template.

x Line 540: "The process of composing a CPA template from two CPP's will often narrow down the amount of ne...." This is quite important and could be highlighted even more. And if the CPA composition cannot get a final CPA, then a CPA negotiation is necessary.

x Line 560: The ebCPP should contain more information how to create an NDD. Also the NDD has to be part of the ebCPP document.

*** Section 7

x Line 565: How to validate a CPA template? Most likely against the CPA schema? But that validation could fail most of the time, if validating the dummy values.

x Line 582: Oops I never thought of having more than one CanSend element for the same "business document"! (As a sidenote it might be interesting to have some validation of the CPP against the underlying collaborative business process, eg if the CPP includes all business documents for example)

x Line 588: The deliveryChannel element would also be an example

x Line 593: This is an interesting problem, it will be very interesting to see how negotiation algorithms deal with such a situation. Might result in a deadlock as noone wants to give in, which could be understandable. Well human to human negotiation might have to finalise the CPA.

x Line 594: The validity of a certificate ... this is interesting as here is a point where a tool also has to "deal" with included XML schemas, so a tool has to understand them as well to know how to get the date filed for example.

x Line 600: In the maintenance, renewal of a CPA, the Start and End elements will be important to negotiate. Hopefully the rest of the elements can stay as they are and only these elements have to be updated.

*** Section 8

x Line 628: What functions?

x Line 644: vanilla SOAP? SOAP without attachments?

x Line 648: HTTP Version 1.1?

x Line 649: A response is synchronious? Line 620 said messaging is allways asynchronous.

x Line 661: Using SSL could go into line 648 to use HTTPS.

x Line 664: "Explanation of NCPA Example"? Isn't the NCPA in Appendix C "THE" NCPA and not only an example? If parties want to build their own NCPA than yes. But the NCPA in Appendix C might be the "DEFAULT" NCPA and not only an example one....

*** Section 9

x Line 676: "... with composing a CPA from two CPP's." Should it not go into the ebXML CPPA Spec, Appendix E (CPA composition) then? -- It might be worth to have a seperate document called the "ebXML CPA Composition", especially when this functionality becomes a real demand.

x Line 700: What is the same BPSS instance Document? same name, same version and same uuid? or a mix of it? which combinations of those attributes are not the same bpss instance document? Neither clear in the CPPA spec. A boolean table would define when two ProcessSpecification elements are ment to be the same BPSS Instance Document.

x Line 708: If both CPP's specify both roles their is the potential to get 2 CPA's.

x Line 715: Again potentially two possible CPA's.

x Line 733/894: Probably like other "optional" elements and attributes the parties first have to agree whether to use it and then to find a value for it. Havent read further yet but is this two step process accounted for in the NDD, does some attribute in the NDD negotiation item indicate this 2 step process? Watch out for any dependencies of such an attribute/element!

Does this only happen if one party has a value for an optional attribute/element and the other party does not have a value? I assume

that if both parties do not have it the do not want it and if both parties have values for such an optional attribute/element they do want it.

x Line 740: maybe say data type of value rather than type of value

x Line 742: might might be written in capital letters "MIGHT"

x Line 746: a new term "CPA-under-construction" introduced here.

x Line 759: Generally each BusinessTransactionCharacteristics element has to be negotiated individually. Is this an idea to have a sort of CPP global BusinessTransactionCharacteristics information?

x Line 846 and 868: Marty, isnt here also the case, that only the information in the CPA template of the party changes but not necessarily in the CPP? If the party would like to add the certificate or trust anchor to the CPP it can do so but also could just add it (certificate or trust anchor) to the CPA template and finally CPA.

x Line 903: Having more than one PartyInfo in one CPP provides the potential for more than one CPA.

x Line 906: Maybe say that PartyId is not negotiable...

x Line 917: This is actually interesting, also in the CPA composition process as there might be CollaborationRoles from different PartyInfos can match. It may say that some content can be negotiate and that the CPA formation just has to make sure that there is at least one matching CollaborationRole in both PartyInfo elements.

x Line 923: "This MUST be validated" probably in the CPA composition and during the CPA negotiation has to be checked that the roles are OK. This has the interesting question how much validation is part of the CPA negotiation.

x Line 930: ... ThisPartyActionBinding and OtherPartyActionBinding have to match and as far as I can tell the name is the only way to find out if two bindings match or not. OK it might be possible to go down the XML path to the business documents and check if they also match...

x Line 942: ... from email correspondance I thought that in a CPA (version 2.0) only one channelId shall be present. In a CPA template more than one can be present and the CPA negotiation makes sure that in the end only one channelId will be present. So it could say that "The Parties can negotiate which delivery channel to use or add one" instead of delivery channels (plural).

x Line 947: From some study I only can agree with instead of modifying a given DeliveryChannel to create a new one. The main problem that I found is to keep track of references (http://www.schlegel.li/ebXML/peecs_presentation/peecs-2003.pdf).

*** Section 10

x General NDD should be part of the CPPA Spec as it is created along a CPP and a CPA template (based on a CPA composition). Probably if this spec gets version 1.0 it might go into the CPPA Spec or the CPA composition (or generally the CPA formation) gets its own specification and the CPPA Appendix E will just reference to that new specification/document.

x Line 981: An NDD is exchanged but it does not say how it is exchanged; if it is exchanged electronically there needs to be some protocol, else it must be exchanged manually. Also this list item is

open for a negotiation of a cpa template and two cpps. maybe a simplification is to only deal with the cpa template case.

x Line 983: It might say an NDD specific for the CPA template.

GENERAL: A notation such as NDD_cpa and NDD_cpp(a) might make clear what kind of NDD the specification talks about...

x Line 988: Maybe make say that a negotiation item only specifies an element/attribute but not its children ... Here is a general problem of element/attribute dependencies. Allowing to negotiate the BusinessTransactionCharacteristics element, or a specific attribute of the element might have the consequences that other elements/attributes have to be negotiated but are not listed in the NDD.

Comment to above -- clarified in Line 1017+

x Line 995: Again the dependencies of element/attributes might cause problems in the NDD. Example what does mean to negotiate over the DeliveryChannel element, does that include to be able to negotiate over its children (and hence Transport element and DocExchange etc).

Comment to above -- clarified in Line 1017+

x Line 1004: Again the NDD for the CPA template -- NDD_cpa.

x Line 1004: Doesn't the CPA template reference the CPPA XML Schema already? So it would not be necessary to be included in the NDD. But the inclusion would be easier to be used.

x Line 1013: Wasn't there somewhere the information, that choices, optional elements/attributes can be retrieved from the CPPA Schema and hence do not have to be listed in the NDD. Actually I would put the choices into the NDD...

x Line 1017: Is a rule necessary such as only the child element(s) but not its attribute(s) are negotiable?

x Line 1017: element/attribute dependency could be a problem for all rules here.

x Line 1027: Maybe move this section one section ahead.

x Line 1054: ... it seems like the NDD for the CPA template is really only used for combining two NDD's. ** In my current CPA composition implementation the composition algorithm writes out that sort of gap list, in particular a conflict list. It seems that those conflicts do not have a place in the NDD.. The sentence beginning in Line 1056 shows that ... if there are irreconcilable conflicts ... the CPA negotiation does not deal with them ... ** From a previous understanding I was rather thinking that the CPA negotiation will try to overcome irreconcilable conflicts ... thinking of that it would probably become very complicated to have algorithms which can deal with all those possible irreconcilable conflicts ...

At the same time ... if party A runs the CPA composition which results in a CPA template, party A uses its NDD for its CPP. Party A could even use a NDD which is not public, so party B has no way to tell what party A exactly did. What I mean is, party A could simply say, all conflicts found in the CPA composition are my NDD. Is that cheating? Party A could run the CPA composition, saying with its NDD and Party B's NDD. The CPA composition could find some irreconcilable conflicts. To go ahead party A could simply add those conflicts to the NDD for the CPA template, saying that it wants to negotiate over those elements... but reading Line 1054, this is not possible as the NDD for the CPA template only includes, what also

Party B includes in its NDD ... and the CPA negotiation could not get started.

Maybe some real world experiences will help how things will be used, for example will it be a practice to set the negotiation item: "/" in the NDD, saying that everything is negotiatable?

So far I understood, that the CPA negotiation should help to overcome serious problems, but it seems that the negotiation rather deals with issues with are likely possible to be solve. Serious, irreconcilable problems seem to be dealt only in human to human ways, so far. ** I think, this is actually a serious simplification then.

x Line 1059: Comment: This is the important part of what is in the NDD and what the NDD is all about. The instance NDD document in the Appendix is very helpful as it shows how an NDD is used.

x Line 1069: I think there should be two XPath expressions, because of the nature of the CPA, which holds two PartyInfo. Very often, there are pairs, one from party A against one from party B. With two XPath expressions the NegotiatableInformationItem can point to those two items which, in the end, both must have the same value. Which value, the one of party A or the one of party B is changed, does not matter after all. Of course it does matter to the party... but not the CPA document.

For example the NegotiationInformationItem in line 3828 wants to negotiate ALL isNonRepduaitionRequired attributes in both PartyInfos for ALL CanSend elements etc. What about the corresponding CanReceive element???

Further if one party specifies which one CanSend element eg CanSend [@name="ABC"] how do you find the corresponding CanReceive? In the prototype implementation I did for the CPA composition there is a little bit of code which tries to figure out which CanSend goes with which CanReceive. Do you expect this work to be done by both parties? What if they get it wrong?

Further does the example in line 3828 assume that somehow the corresponding CanReceive element will be set accordingly to whatever is negotiated for the CanSend element ... as there must be a matching pair of CanSend and CanReceive.

The same is true for example in line 3836.

Further I dont understand example in line 3846. There is only one value for @syncReplyMode: signalsOnly. Questions:

1) This is probably the same value as is in the CPP? If so why add it to the NDD? Actually in which CPP? In the PartyInfo down somewhere of which party? Does it matter?

2) Does it make sense after all? We shall negotiated that it (attribute) is present and that the value is signalsOnly? If both NDDs have been used to create this NDD then there is not need to negotiate about it and the attribute can be set to "signalsOnly" without negotiation.

3) This shows the difficulty to track dependencies. If this attribute changes from a previous value to "signalsOnly" this would mean that things in the CPA template get out of validity because of this change. Unfortunately I still have difficulties to understand the syncReplyMode attribute but I think a change from none to signalsOnly will have consequences of having additional elements/attributes within both PartyInfo elements. Big question: Who is responsible that those dependencies are met? 1st question: what dependencies are they?

2nd question: Are those elements negotiatiable? Probably yes and they have to be "added". What are the preferences of those new elements/attributes? Those preference values are not listed in this currrent NDD ... Does this result in an irreconcilable conflict?

Sorry for not providing the answers but only questions :)

x Line 1073: Unfortunately I do not have any experiences if the following list covers all possible scenarios.

x Line 1077,1080,1082: Is a data type necessary or should it be looked up in the CPA Schema?

x Line 1093: Do you mean if the element/attribute is present or not or the value is present or not? If the value is not present, does that mean that the element/attribute can be removed from the CPA? Maybe there is there another type necessary: the type which negotiates whether an attribute or element must be present or not and then if present the value has to be negotiated.

*** Section 11

x Line 1212: maybe forward link to section 11.4 and its error codes.

x Line 1217: The cpaID of the CPA template is negotiatiable (see line 879) so the CPATemplateID element value can change throughout a negotiation ... Does this matter?

x Line 1249, 1265: InitiatingParty and RespondingParty stay the same even after the role change, which is OK.

x Line 1238, 1261, 1279, 1262, 1279, 1300: Change all Uri to lower case uri ... or all to uper case URI or just make it consistent.

General question when dealing with uri's: Does this mean that the parties must have a webserver where the document (NCPA in this case) can be accessed by the other party? This would most likely be outside the Messaging System Interface (MSI), and maybe even the Business Service Interface (BSI). Is this generally no problem to provide a HTTP access point to the outside world? Does the negotiation software have the priviliges to change the content of that URI? Does it even have the security permissions to provide a http access point?

x Line 1287: General question: I think there was the discussion about "CPA Offer Doc", "CPA_Offer_Doc" and "CPAOfferDoc". Is it necessary to make it consistent to one version only?

x Line 1310: Ohhh OK so the CPA template HAS to be updated by each party based on what the negotiation content was. This is actually good, I think, so each party can check verify that the CPA template is updated the same way by each party; and this check can be made after each counter offer.

x Line 1316: This is actually the important element here!

x Line 1318: Marty, I think we once discussed, that not the CPP or NDD do get changed but just their information in the CPA template ... e.g. the PartyInfo element in the CPA template looks slightly different to the PartyInfo of the initial CPP. It will be up to the party to update the "public" CPP or not.

x Line 1331: "Any item that has been deleted by one party can no longer be re-inserted in future counter offers" is actually an

important rule and should maybe go to a negotiation rules section.

x Line 1325: What is the reason to have a party list only its accepted/updated, deleted or inserted items? It might be helpful to have the full history of the negotiation in each message, but that is not really necessary as both parties maybe already have such a history. But having the history in each negotiation message the state of the last CPA template could be created by applying the changes (update, add, delete, accept) to the initial CPA template.

General: Maybe the specification could say, that with the initial CPA template and the history of the negotiation it shall always be possible to get to the last CPA (or last CPA template) version at any stage.

x Line 1338: Maybe, as mentioned above, there needs to be two XPath expressions for each item.

x Line 1338: Another point is that the XPath expressions might become wrong after inserting new elements etc. I think the "old" accept items XPath expressions should not be updated, so that one follows the history of the negotiation (add x, update x, update x, add y, update y, accept x, etc) with the XPath expression at that time is possible.

x Line 1352: Is the Original value redundant? It could be traced back to a) the CPA template or the InsertItem. But maybe its just easier to keep the original value close.

x Line 1346: Does "during all exchanges prior to this Message" contradict line 1326 "... (including the ones being accepted by this Message)" ?

x Line 1360: Yes I agree as I dont understand that. Could it be that the required or preferred can be found by looking at the NDD NegotiationInformationItem type? Do not understand this one.

x Line 1395: Seems to make it more difficult to create a good NDD...

x Line 1405: Maybe add a sample NDD to the initial offer in figure (1).

x Line 1412: The Responding Party did summarise all "AcceptedItem" in the XPath expression ... is this valid? According to line 1346 or 1326 it should include the previous accepted items as well ...

x Line 1412: Also I am not sure if the "AcceptedItem xpath="/ElementB[0]" complies with line 1346 or 1326 ... the initiating party only has to list accepted by the sending party as ElementB[0] was accepted by the responding party it should not be listed in the first counter offer of the initiating party ...

x Line 1419: ... does the party already now what it will change at that stage? Or does the party just say, hey I am sending you a new counter offer? OK Line 1420 says "might" ... in that case the receiving party of a "CounterOfferPending" document must check the NegotiationContent element! But maybe it would make it easier to have an empty NegotiationContent element in a non "InitialOffer" and "CounterOffer" document.

x Line 1421,22,23,24,25,26,27,28: does not make sense to me. I thought those are only in a "CounterOffer" document ... In Particular rejection should be in a "Rejected" message type ... not in a "CounterOfferPending" message.

x Line 1441: Even if it the CPA template id is negotiated?

x Line 1483: Type "OurOf..." -> "OutOf..."

x Line 1483, 1486: Why the need for a "_"?

x Line 1514: The CPA must be a valid XML instance document. Do the parties check that prior to sending the final documents?

*** Section 12

x Line 1516: The "Negotiation Protocol" section could maybe moved to the beginning of the Specification ...

x Line 1608: Is accepted necessary?

x Line 1653: General: Maybe add somewhere, that all NegotiationInformationItems must be accepted before the negotiation can end, ... before a party can accept the offer/counter offer. and otherwise if there are still open issues and one party accepts the offer/counter offer an error must occur.

x Line 1662: Ooops here it says if one party sends an accept message it, implicitly accepts all remaining open items. Is that good? Also is it possible after all, if there is a range value for example? Which value would you take if the NDD provides a list of values?

x Line 1777: Maybe add why two binary collaborations are used (because the role changed could not be modelled by one binary collaboration). Still not sure why you wanted to have different roles?

x Line 1787: the first "to" is to much I think.

x Line 1822: Type and confusion with "_" : Change from "CPA_Negotiation_CounterOfferInitiator_Role" to "CPA Negotiation Counter Offer Initiator"

x Line 1823: see above

x Line 1782: "Offer-Counter-Offer Choreography" and

x Line 1859: "Final CPA exchange" not reviewed as they are simply rephrasing the BPSS Instance document and its time for dinner ;)

*** Section 13

x The negotiation algorithm is probably the hardest part. Any idea how these algorithms will evolve? Any idea who will implement these algorithms? Any ideas on what background knowledge these algorithms will be based upon? Any one started with a study of what the requirements for a negotiation algorithm are? Any thoughts of having another specification/document/report on ebXML Automated Negotiation of CPA Negotiation Algorithm Document or is this fully open to implementers.

x General: Mentioned before ... if any Negotiation algorithm goes in here it goes under the IBM patent, which will deal with version 1.0 of this spec, right? Is that OK?

Hope this list helps to advance the 0.10 version. I think some testing with sample CPP's, sample CPA templates and sample NDD's would reveal much of what is missing and what is needed. This is specially true for sections 10 (in particular 10.2) and section 11 (in particular 11.1.9). NDD probably need some further thinking ...

Kind regards

Sacha Schlegel