Possible new work items for CPPA TC

Martin W. Sachs
Tony Weida
Other Team Members

This is a list of possible changes and enhancements to the CPP-CPA specification.

The sources of these topics are the TP team listserver, various meetings, and other suggestions. Within each section, topics are not listed in any particular order.

NOTE: This document is an overview. More discussion of these topics is in "CPA-CPP Changes to Consider", 5/9/01, by Marty Sachs, previously distributed, and in various recent discussions on the ebXML-TP and ebxml-CPPA listservers.

The suggestions are grouped in these categories:

- New function
- Loose ends from Version 1.0.
- Longer Term Enhancements
- Mechanics of the Specificaton

1 New Function

1.1 Security definitions

- Technical Architecture Risk Assessment technical report recommendations related to CPP/CPA. (Note: some items below may duplicate material in that report.)
- Security profile developed by the ebXML security team.
- We should consider a Security policy element.
- Public-Key Infrastructure issues.
- Improvements to packaging definition including security capabilities. Specific problems related to XMLDSIG have been noted (Dale Moberg 5/3/01)
- Nonrepudiation improvements including possible addition of other elements that reflect choices that can be made (Transform?). A possibility is that this element could take the form of a Signature "template" which effectively provides all the requisite binding information including reference URI(s) with only the Digest and actual signature omitted. This would be similar to the way we now define the signature. See listserver discussion 6/13-01 - 6/14/01.
- Specification of nonrepudiation of receipt.
- Signing of payload and header vs. signing only of payload and response.
- Certificates: replace ds:keyinfo element by a definition that does not embed the actual ceritficate in the CPP or CPA.
- Define security attributes under the Characteristics element in enough detail to understand what has to be specified in doc exchange and transport to support them and enable a tool to check for consistency between Characteristics and the details in DocExchange and Transport.
- Use of third-party security services (this is a special case of an intermediary).

1.2 Negotiation

- CPA contents
- Business-level parameters

1.3 CPP and CPA tools

This topic consists of non-normative discussion of the CPP and CPA tools as an addition to the CPA-composition discussion in appendix F of the Ver. 1.0 CPP-CPA specification.

- CPP composition tool
- CPA composition tool
- CPA digest tool

The result of this work might be a technical report. This might also be viewed as part of the Negotiation work item.

1.4 Provision for alternative business-collaboration specifications

We should provide for use of "foreign" business-collaboration specifications as alternatives to the Specification Schema model. Examples might be:

- Hand-crafted collaboration protocols based on a tpaML-like language
- Collaboration protocols based on WSDL
- Collaboration protocols based on alternative models such as WSFL, BPML, or whatever these evolve into.

Some of these may involve joint work with the BP team or collaboration with the appropriate W3C team.

1.5 Middleware interoperability

- Upper interface of Message Service
- Interface between middleware and bridge to legacy applications
- Anything else to support interoperability aspects of CPA and middleware?

Probably this should be joint work among CPPA, MSG, and BP teams. Perhaps some new OASIS team should be created to lead this work.

1.6 Business-level timeouts

Should the CPA provide for specifying timeout, number of retries, and retry interval for business-level responses? If the Specification Schema provides these parameters, then they probably have to be given values in the CPA. As with the security attributes, what is in the Process Specification document can be viewed as a default or recommendation with the agreed values specified in the CPP/CPA. For example, the timeout might depend on a Party's specific implementation of the process.

1.7 Support for alternative message services

The specification tries to make it clear that a user of a CPP or CPA may use an alternative message service such as SOAP or XML Protocol. However, the specification does not prescribe a formal way to add the alternative messaging service. The user must revise the schema or DTD to eliminate the ebXMLBinding element and add whatever new element is needed. For this approach, we need to change the cardinality of ***ebXMLBinding*** to (0 or 1). Other possibilities:

- ♦ Add an extensibility element to be used when introducing an alternative messaging service.
- ♦ Provide xxxBinding elements for commonly used messaging services. SOAP 1.1 and XML Protocol (when available). All these "supported" elements would be defined as part of an enumeration (1 out of the list would be required).

1.8 Intermediaries and Multihop Scenarios

Use of intermediaries may need to be accounted for in the CPA. Intermediaries include trading services of various kinds. The use of a proxy outside a Party's firewall is a specific case of an intermediary.

1.9 TPA reference element

It has been proposed to add an optional element to the CPA that provides a reference to an associated "traditional" contract or TPA. This should be able to be either a text string or the URI of an electronic (e.g. XML) representation of the contract or TPA. It should be stated this element is for information only; its presence or absence is independent of whether a contract does or does not exist.

It will be necessary to consider the relation of this element to the isLegallyBinding attribute in the Process Specification document. (See listserver discussion June 26-27, 2001). It appears that isLegallyBinding is used only to flag an instance of a business transaction as for test purposes only, in which case there is probably no relationship.

1.10 Specialization of the Process-Specification Document for specific pairs of partners

Specialization of the Process-Specification document for specific pairs of partners can be done using the 'substitution' capability that was added to BPSS at Vienna. In general this concerns how to have generic business processes, and yet be able to process specializations of those for the specific partners. This would be joint work with the CPPA and BP teams.

1.11 Composition of Services

David Burdett asked whether it is possible to use the same service in different business processes. An example is a payment authority that offers a payment authorization service that accepts a payment request and returns a payment response that conforms to some part of the IFX specification. This could be used in many different processes to make a payment, e.g. to pay an invoice, to get foreign exchange, etc. It would be really good if a payment authority could just define the service once and then everyone could use it in whatever business process it is needed in. This can be done with the existing CPA definition since a CPA can reference multiple Process Specification documents, one of which could be the payment process. However, there is no way to choreograph the interaction between the payment process and the accompanying business process. This is probably a BPSS issue. The IBM WSFL web-services proposal includes composition of services from simpler services.

2 Miscellaneous Details and Loose Ends

2.1 Interfacing the CPA/CPP to the Specification Schema XML document

- Have the CPA directly select the binary collaboration(s) that are applicable. Add to the ***ProcessSpecification*** element a child element (cardinality one or more) that contains an xlink pointing directly to a binary collaboration
- Currently, there is one ***Characteristics*** element per delivery channel. Yet each business transaction may have a different combination of characteristics. We need a better way of specifying characteristics than by multiplying the number of delivery channels. See "alternatives and choices" below.
- Currently, the link from the ***action*** attribute (***Override*** element) to the matching business transaction in the Process-Specification document is the equality of the value of the ***action*** attribute to the value of the ***name*** attribute in the desired ***BusinessTransaction*** element. Use of an xlink may be better for the installation tools.

2.2 Normative Appendix on Use of the CPA with the ebXML Message Service

This appendix is an outstanding item from Ver. 1.0. There are ambiguities in the Message Service Specification that result from treating the CPA as optional. Therefore, the CPP-CPA specification has to define exactly how various elements in the message header are to be used with a CPA. This item should be joint work with the MSG team. Some examples:

- Possible clarification of the role of the Service and Action elements in the header.
- How the RefToMessageId element is to be used in application-level request and response messages (see "Routing of Response Messages" below).
- Clarification of some of the Reliable Messaging issues pointed out by Arvola Chan (see "Reliable Messaging Consistency" below).
- List and discussion of all elements in the message header that relate to the CPA.

2.3 Reliable Messaging Consistency with Message Service Specification

Arvola Chan (posting of 7/15/01) reported a number of problems with the Reliable Messaging definition in the Message Service Specification. Fixing some of these may require coordination between the CPPA and MSG teams. For example, he questioned whether TimeAccuracy and TimeToLive should be added to the CPA.

2.4 Routing of Response Messages to the Correct Software Entry Point

- The specification should be reviewed for clarity in the definitions that determine how to route a reply message to the correct software entry point at the recipient of the reply message. Can the installation tools handle these definitions? In most cases it should be sufficient to route the message by using the Service and Action elements in the message header combined with which role the party receiving the message plays. The following should be checked for accuracy and clarity: 7.55 Role element, 7.5.5.1 name attribute, 7.5.7 Service element, and 7.5.8.1 action attribute. See below regarding the RefToMessageId element in the message header.

- There appears to be an ambiguity for the following case. The ambiguity should be confirmed or refuted and, if it is ambiguous, either fix the ambiguity or put in a statement that this is not a valid case. The case boils down to the same party playing both roles in the same business transaction.
  - Each Party to the CPA includes two CollaborationRole elements that point to the same Process-specification document.
    - In one CollaborationRole element, Party A has the role "seller" (for example) and Party B has the role "buyer".
    - In the other CollaborationRole element, Party B has the role "seller" and party A has the role "buyer".
  - The same combination of binary collaboration and business transaction is performed in both of the above cases (i.e. the two Parties can switch roles).
  - Both CollaborationRole elements specify delivery channels with the same transport address (e.g. URL).
  - The message service cannot tell whether an arriving message is a request or a response message. It can only route based on transport address, service, action, and role. If the same delivery channel (same transport address) is used with both CollaborationRole elements, the messaging service cannot tell whether to route the message to Party A as "seller" (the first CollaborationRole element) or to Party B as "seller" (the second CollaborationRole element).
  - The solution is to make use of the RefToMessageId element in the message header as part of the routing information. See the "RefToMessageId" below.

2.5 RefToMessageId element in Message Header

The following requires some definition work with the MSG team.

There is no explicit statement in the Message Service spec about the use of the RefToMessageId element in messages other than message service to message service control messages. Use in application-level messages is valuable and will not interfere with the currently defined use and the necessary words should be added.

- In an application-level response message, the RefToMessageId element should contain the ID of the message that the message is responding to. This is necessary to disambiguate the case described in "Routing of Response Messages" above.
- At the same time, words should be added which allow the RefToMessageId to be used in an application-level request message. This would, for example, allowing a message requesting a compensation action to point to the message being compensated.
- It may be necessary to add an indicator somewhere in the message header that the message is a response to a prior application-level message. This indicator would be supplied by the sending application and would enable the receiving system to recognize that the message is a response and the RefToMessageId element should be used as part of the information that routes the message to the appropriate software entry point.

2.6 Maximum Lifetime of a Conversation

There may be a need for an element in the CPA that specifies the maximum time for a conversation to live. This would detect hung conversations where neither party is sending a message or waiting for a response. The timeToPerform attribute of the binaryCollaboration element in the Specification Schema document may do the job. It will have to be linked to an element in the CPP/CPA.

2.7 Sending Protocol

There is a question of whether the ***SendingProtocol*** element is sufficient and in the correct place in the CPP/CPA. One question is whether there is also a need to specify send capabilities in the document-exchange section. Another question is whether the send capabilities should be specified within the delivery channel at all. One possibility is to specify send capabilities at a higher level (e.g. a sibling element to the DeliveryChannel element) so that each set of send capabilities can be referenced by more than one delivery channel.

Is the ***SendingProtocol*** element needed in the CPA at all? Currently, its cardinality in the CPA is "required". It was intended as an assist to a CPA-composition tool. If it is needed in the CPA, is a cardinality of "Required" correct?

2.8 Receiving Protocol

Should the ***Endpoint*** and ***TransportSecurity*** elements be changed to be child elements of ***ReceivingProtocol***? Currently, they are siblings of ***ReceivingProtocol*** and the text states that they apply to ***ReceivingProtocol***.

2.9 Transport Security

Transport security is specified under receive properties (i.e. in the delivery channel) but the spec states that it applies to messages in both directions. That means that the other party's delivery channel must specify the same transport security definition. This may be a problem for a CPA composition tool. Is it necessary for the same transport security definition to apply to messages in both directions? Couldn't the transport security properties be different for messages sent by each party? A better idea may be to put transport security also under sending protocol and delete the statement about applying to messages in both directions.

2.10 Digital Envelope

There is no counterpart of the CPP/CPA Digital Envelope element in the message service specification. Discussion on the listserver suggests that the question is whether Digital Envelope encryption is to be performed by the messaging service or by a level above the messaging service. If it is to be performed by a level above the messaging service, it is reasonable to specify it in the CPA even though the Messaging Service doesn't perform the function.

Is Digital Envelope needed at all?

2.11 Alternatives and Choices

We need an efficient way in the CPP to describe alternatives and priorities among choices. Using multiple delivery channels may not be the best approach since if there has to be a unique delivery channel for each combination of alternatives, the number of delivery channels can explode. One approach is to define alternatives and choices in a way similar to certificates and packaging, as elements that can be referenced from delivery channels.

Examples:

- Delivery channel Characteristics element
- Message security definition

2.12 XMLDsig approval status

As XML Digital Signature advances in approval status, it will be necessary to update the text and examples of ds:Signature and ds:Reference (under ProcessSpecification) if the XMLDsig specification changes in any significant way. It will also be necessary to update the URL in reference to the XMLDsig specification.

2.13 XML Schema approval status

Now that XML Schema has achieved Recommendation status, it will be necessary to update the schema file and some text to account for changes such as certain data types that are used in the CPP/CPA specification. It may also be necessary to update the URLs in some examples and in the reference to the specification. These changes should probably not be made until the XML Schema tools are updated to Recommendation status.

2.14 Exploitation of XML Schema

Consideration should be given to exploit XML Schema to provide function that isn't defined in the DTD. Once this is done, the DTD should probably be eliminated since it won't necessary be able to capture the advanced Schema functions.

2.15 FTP

The FTP definition may need further elaboration. For example do the Parties to a CPA need to agree on:

- Transfer type (binary or character)?
- Password properties?
- Is PUT the correct operation for receiving messages?
  - ♦ Note: In the CPP, the delivery channel specifies RECEIVE properties
- GET as well as PUT?
- Passive mode (yes or no)?
- Control port number for passive mode?
- Anything else with regard to firewalls?
- Anything else?

2.16 PartyId definition and example

Both examples of PartyId in ver. 1.0 have the type attribute included although according to the text, all the information is in the value of the attribute. We need to formulate an example in which the type attribute is essential.

2.17 PartyId type

It has been suggested that a negotiation of PartyId type may be desirable since a given Party may not be capable of interpreting all possible PartyId types. One possibility is to add an element by which a Party can indicate which PartyId types it understands.

2.18 Digests of Other External Documents

If other external documents, such as security profiles, are introduced, the possibility of creating digests of those document, similar to what is specified for the Process Specification document should be considered in order to detect alterations.

2.19 Publishing Party capabilities with a CPA template instead of a CPP

There was discussion on the list server Feb. 6-8 about use of CPA Templates in place of CPPs. This would simplify things for a small business so that it doesn't have to go through the whole CPP and composition process when all he needs to do is fill in a few items in a CPA prepared large business. The words in the spec are sufficiently permissive to allow the possibility of the use of CPA Templates but one could make it more explicit

2.20 XSD and DTD Errors

Several errors have been found in the Ver. 1 XML Schema for the CPP-CPA specification. I suggest posting the corrected schema to the CPPA list and putting the errors on the list for the maintenance release. Known errors:

- Several violations of the Schema specification were pointed out by Michael Wang on CPPA listserver 6/26/01.
- Extraneous Type element in both the DTD and the XSD. This is apparently the Type element that was a child of CPAType and wasn't deleted from the DTD and XSD when we removed CPAType.

2.21 Errors in CPA example

Yukinori Saito (5/16 and 5/17/01) pointed out errors and suggesting corrections to the CPA sample regarding incorrect use of ID attribute "N08". This could be corrected and distributed on the CPPA listserver until we issue a maintenance release.

2.22 Post-publication Quality Review Comments

Sun submitted these quality-review comments after the version-1 specification was published.

- It is inappropriate to include the line "generated by XML Authority" in a normative DTD.
- The title of Appendix D should be "W3C XML Schema Document" or "XSD Schema

Document instead of "XML Schema Document". Note that the title of the schema specification is "XML Schema".

3 Longer Term Enhancements

3.1 Interaction between configuration inside a Party and the CPA

In general, configuration matters are internal to each party and should not appear in the CPA. However there may be CPA implications, especially if internal configuration information overrides fields in the CPA. If that can happen, it needs to be documented in the CPA specification.

3.2 CPA between more than two parties

Extension of the CPA to more than two parties could be considered.

3.3 Additional Transport Protocols

Consider adding support for additional transport protocols such as:

- IIOP
- EDI value-added networks

3.4 Payload Compression

Should the CPP and CPA support payload compression? This element would indicate whether the sending party is sending compressed payload and what the compression algorithm is. It could be:

- Once for each party's set of business transactions
- Once per message definition.

3.5 Include higher-level abstractions in the CPA

There has been a suggestion to extend the CPA to include higher level abstractions like service information and contractual obligations.

4 Mechanics of the Specification

4.1 Figures

The figures should be redrawn with Word's own drawing tool. This may allow better control over the positioning of the figures than is true with the current figures imported from PowerPoint. The positions of the current figures are notoriously unstable with respect to nearby text changes.

Redrawing the figures with the Word drawing tool should also allow automatically numbered captions to be used instead of the captions currently drawn within the figures.

4.2 Definitions of Terms

If the post-Vienna disposition of the ebXML specifications renders global documents, such as the ebXML glossary, inoperative, the definitions of terms should be restored to the CPA-CPP specification. The definitions in the TP Requirements document are a starting point but this list will have to be updated.

4.3 Publication of text forms of the DTD and XSD files

We need to decide where to publish the text forms of future versions of the specification.