CPA-CPP Changes to Consider

Martin W. Sachs

This is a list of possible changes and enhancements to the CPA-CPP specification. All should be considered beyond ver. 1.0 (i.e. "post-Vienna").

The sources of these topics are the TP team listserver, various meetings, suggestions, and some of my own thoughts. Section headings approximate classifications. Within each section, topics are not listed in any particular order.

1 Near-term improvements

1.1 Security definitions

See the ebXML Technical Architecture Risk Assessment technical report for recommendations related to CPP/CPA. (Note: some items below may duplicate material in that report.)

We need to incorporate the security profile developed by the ebXML security team.

We should consider a Security policy element.

Public-Key Infrastructure issues.

Post-Vienna Packaging Issues:

- Investigate using "grammars" to provide more compact means of expressing packaging parsing and generative capabilities.
- Enhance notation to capture the "virtual packaging" used by XMLDsig external references.
- Coordinate or extend Packaging to include enhancements to descriptions of security capabilities that impact interoperability that have been raised by the Security subgroup, including Trust Anchors and other security policy and profile information.

We will need to scrub the existing security elements in the specification to conform to what the security team has developed. The current security elements are mostly taken unchanged from the IBM tpaML proposal.

Depending on how it is used, the ds:keyinfo element can contain an actual base-64-encoded certificate. Except for signing of the CPP or CPA, there is no need for actual certificates to be embedded in the CPP or CPA. The Certificate element should be changed to point to information at a key-management service instead.

Some of the security attributes under the Characteristics element put requirements on the security definitions in the doc exchange and transport sections. Those attributes have to be defined in enough detail to understand what has to be specified in doc exchange and transport to support them.

Discussion is needed on the function of nonrepudiation in the delivery channel *vs.* the certificate in the role tag. Chris Ferris' comment on this:

> I for one believe that it is only useful when signing both header and payload together. Of course, there has also been discussion that the only meaningful NR signature is that where the "application" signs the payload (or header and payload) and the ack. This needs more scrutiny and discussion with the ta-security folk.

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

This includes:

- Currently, the installation tools may have problems finding the desired business transactions in the Process-Specification document since they must search for all business transactions with the appropriate authorized roles. This is complicated by the fact that more than one binary collaboration may have authorized roles corresponding to the parties to the CPA. An improvement is to add to the ***ProcessSpecification*** element a child element (cardinality one or more) that contains an xlink pointing directly to a binary collaboration. If more than one binary collaboration has roles matching the parties to the CPA, more than one such element could be included. Thus the CPA would select exactly the binary collaboration or binary collaborations desired. The tools could then find them and verify that their authorized roles match the parties to the CPA. The ***Role*** element would still be needed because its job is to name the authorized roles that correspond to the parties to the CPA.
- Currently, there is one ***Characteristics*** element per delivery channel. Yet each business transaction may have a different combination of characteristics. Relating the attributes of ***Characteristics*** to individual business transactions can be done by defining sufficient different delivery channels and using the ***ServiceBinding*** and ***Override*** elements to associate sets of characteristics with different business transactions. Is there a better way?
- More detail is needed in the definitions of the security attributes of ***Characteristics*** in order to determine whether the specification of the delivery channel includes the corresponding functions. The specification should permit CPP and CPA tools to detect inconsistencies.
- 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. An installation tool has to search the Process-Specification document for the ***name*** attribute with the matching value. An alternative is to encode the ***action*** attribute as an xlink that points to the element by referencing the value of its ID attribute. If the xlink alternative is an improvement for the installation tools, consider making the change.

1.3 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, such as when one party's application has been terminated by a fatal error. Middleware could use this lifetime specification to terminate hung applications. It should be stated that the maximum time for a conversation should exceed the sum of the business-level response times for the individual business transactions in the conversation. The timeToPerform attribute on 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.

1.4 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? It was intended as an assist to a CPA-composition tool. If it is needed in the CPA, is a cardinality of "Required" correct?

1.5 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***.

1.6 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. A better idea may be to put transport security also under sending protocol and delete the statement about applying to messages in both directions. The two parties' transport security definitions still have to match but putting transport security in sending properties may assist 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?

1.7 Alternatives and Choices

We need an efficient way in the CPP to describe alternatives and priorities among choices. Using multiple delivery channels isn't enough since if there has to be a unique delivery channel for each combination of alternatives, the number of delivery channels can explode.

1.8 Appendix on Mapping of CPA constructs to ebXML Message Header

This appendix needs to be written.

1.9 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.MWS. For example, the timeout might depend on a Party's specific implementation of the process.

1.10 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.

1.11 XML Schema approval status

Now that XML Schema has achieved Recommandation 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.

1.12 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.

1.13 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?

It has been pointed out that those FTP characteristics beyond the basic FTP PUT specification that is in our specification are very vendor-dependent and not necessary controlled by the user. However this is a strong argument for specifying those characteristics in the CPP and CPA to provide additional function while enabling a CPA-composition tool to detect incompatibilities.

1.14 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. A CPA composition/negotiation tool can then use this information in conjunction with the PartyId type each Party states for its own PartyId to select acceptable PartyId types for the two Parties.

1.15 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.

1.16 Support for alternative messaging services

The specification tries to make it clear that a user of a CPP or CPA may use an alternative messaging services 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. This makes such an implementation non-compliant with the CPP/CPA specification. A way is needed to formalize the use of an alternative messaging service while still being compliant with the specification. Components of an approach include:

- Change the cardinality of ***ebXMLBinding*** to (0 or 1).
- Either:
  - 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).

1.17 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.

We need to consider multihop scenarios in which, for example, an intermediary is in between the two Parties. In some scenarios, the intermediary should be able to be invisible in the CPA between the two endpoints and each endpoint Party has its own CPA with the intermediary covering the services the intermediary provides. In other cases, it is not clear that the intermediary can be totally invisible. There are also security issues with intermediaries.

It has been suggested that support for the ebXML Message Service's ***Via*** element, which relates to intermediaries, is needed in the CPA.

2 Longer Term Enhancements

2.1 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:

- Collaboration protocols based on alternative models.
- Hand-crafted collaboration protocols based on a tpaML-like language
- Collaboration protocols based on WSDL

One problem with this is that the linkages between the CPP/CPA and the collaboration protocol document assume the structure defined by the Specification Schema model. The alternative model might not fit this structure. For example, problems could arise with the ***action*** attribute of the Override element and the attributes of the delivery channel characteristics element. Possible solutions include:

- Defining the grammar for specific alternative business-collaboration specifications.
- Providing an extensibility element to permit custom design of the grammar.

2.2 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.

2.3 CPA between more than two parties

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

2.4 Additional Transport Protocols

Consider adding support for additional transport protocols such as:

- IIOP
- EDI Value-added networks

2.5 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 Mechanics of the Specification

3.1 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.

3.2 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.